SIAM J. Applied Dynamical Systems
Vol. 22, No. 4, pp. 2890–2926
© 2023 Society for Industrial and Applied Mathematics

Regression-Based Projection for Learning Mori–Zwanzig Operators*

Yen Ting Lin†, Yifeng Tian†, Danny Perez‡, and Daniel Livescu§

Abstract. We propose to adopt statistical regression as the projection operator to enable data-driven learning of the operators in the Mori–Zwanzig formalism. We present a principled method to extract the Markov and memory operators for any regression models. We show that the choice of linear regression results in a recently proposed data-driven learning algorithm based on Mori's projection operator, which is a higher-order approximate Koopman learning method. We show that more expressive nonlinear regression models naturally fill in the gap between the highly idealized and computationally efficient Mori's projection operator and the most optimal yet computationally infeasible Zwanzig's projection operator. We performed numerical experiments and extracted the operators for an array of regression-based projections, including linear, polynomial, spline, and neural network–based regressions, showing a progressive improvement as the complexity of the regression model increased. Our proposition provides a general framework to extract memory-dependent corrections and can be readily applied to an array of data-driven learning methods for stationary dynamical systems in the literature.

Key words. Mori–Zwanzig formalism, Koopman representation, nonlinear projection operators, data-driven learning, regression, neural networks

MSC codes. 37M99, 46N55, 65P99, 82C31, 37M05

DOI. 10.1137/22M1506146

1. Introduction.

More than half a century ago, Mori [33] and Zwanzig [54] developed a mathematically exact formalism for constructing reduced-order models for dynamical systems in nonequilibrium statistical mechanics. In this context, the typical degrees of the freedom of a system are of the order of Avogadros number. In contrast to tracking the dynamics of each degree of freedom, reduced-order models (also referred to as the coarse-grained models in the context of statistical physics) are developed to describe the dynamics of a relatively small

*Received by the editors June 29, 2022; accepted for publication (in revised form) by G. Gottwald May 22, 2023; published electronically October 13, 2023.
https://doi.org/10.1137/22M1506146
Funding: The work of the first author was primarily supported by the Laboratory Directed Research and Development (LDRD) project "Uncertainty Quantification for Robust Machine Learning" (20210043DR). The work of the first, second, and third authors was partially supported by the LDRD project "Accelerated Dynamics Across Computational and Physical Scales" (20220063DR). The work of the fourth author was partially supported by the LDRD project "MASS-APP: Multi-physics Adaptive Scalable Simulator for Applications in Plasma Physics" (20220104DR).
†Information Sciences Group, Computer, Computational and Statistical Sciences Division CCS-3, Los Alamos National Laboratory, Los Alamos, NM 87545 USA. These authors contributed equally. (yentingl@lanl.gov, yifengtian@lanl.gov).
‡Physics and Chemistry of Materials Group, Theoretical Division T-1, Los Alamos National Laboratory, Los Alamos, NM 87545 USA (danny_perez@lanl.gov).
§Computational Physics and Methods Group, Computer, Computational and Statistical Sciences Division CCS-2, Los Alamos National Laboratory, Los Alamos, NM 87545 USA (livescu@lanl.gov).

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

number of variables of interest.[1] In the jargon of model coarse-graining, this set of dynamic variables is referred to as the "resolved" variables, and the remaining degrees of freedom in the system are referred to as the "unresolved" ones. Reduced-order models are particularly useful in describing the emergent phenomena at the mesoscopic or macroscopic scales. They are also more computationally efficient to simulate (in comparison to simulating the full system) and thus play an essential role in bridging multiscale models.

The key difficulty in constructing reduced-order models is the *closure problem*: one must quantify the effect of the unresolved variables before self-contained evolutionary equations of the resolved variables can be prescribed. To solve the closure problem, Mori and Zwanzig adopted an elegant approach which uses functional projections. Formally, a prespecified projection operator is used to map any function which depends on both resolved and unresolved variables to a function that only depends on the resolved variables. As the latter does not depend on the unresolved information, one can then prescribe a set of evolutionary equations for the resolved variables. The dynamical system in terms of only the projected functions is of course only an approximation of the true dynamics. The Mori–Zwanzig (MZ) formalism further quantifies the error of the approximation and derives how the error would propagate into the future by the dynamics. The central result of the MZ formalism is the generalized Langevin equation (GLE), which stipulates that the evolutionary equations of the resolved variables depend not only on their instantaneous values but also on their history. As a mathematically exact formula, the memory dependence, which depends critically on the choice of the resolved variables and the projection operator, quantifies the interactions between the resolved and unresolved degrees of freedom.

Theoretically speaking, there are infinitely many choices of the projection operators. In the literature, the commonly chosen projections include (1) Mori's linear projection [33], (2) the finite-rank projection operator [6], which is Mori's projection with a set of orthonormal basis functions, (3) projection by assigning unresolved variables to zero [38, 39, 40, 41], and (4) Zwanzig's nonlinear version of projection [6, 54]. It is generally difficult to derive analytic expressions for the MZ formalism. Several recent studies [24, 29, 31, 32, 34, 37, 49] have established that with Mori's linear projection operator, it is possible to adopt a data-driven approach to learn the MZ operators using the time series of the resolved dynamics. In addition, it was shown [24] that these methods provide higher-order and memory-dependent corrections to existing data-driven learning of the approximate Koopman operators [43, 44, 45, 52].

In spite of being a consistent theoretical framework to the approximate Koopman learning, Mori's projection operator is greatly limited by its linear nature. That is, after the projection, all functions must be expressed as linear combinations of a set of a priori specified resolved variables. The quality of the prediction thus solely depends on the choice of the resolved variables. To the authors' best knowledge, there is no principled way to select an optimal set of resolved variables for general dynamical systems. With a nonoptimal choice, the difference between the projected dynamics and the true dynamics can be unsatisfactorily large [24].

[1]The term "reduced-order models" could mean different things in different contexts. We adhere to the definition that originated from nonequilibrium statistical mechanics, in which the formalism was established. With this definition, "reduced-order models" is interchangeable with the terms "partially observed" and "underresolved" dynamical systems.

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

The same problem also manifests itself in the approximate Koopman learning framework [52]. Furthermore, Mori's projection operator is not compatible with many nonlinear closure schemes [10, 30]. Neither can we generalize the Mori's restrictive linear projection operator to those data-driven learning methods based on modern machine learning [5, 22, 27, 51], with which one relies on the neural networks to identify the nonlinear relationship between the input (what we know at the present) and the output (what we want to predict in the future).

At the other end of the spectrum is the Zwanzig's construct [6, 54], which uses conditional expectation as a projection operator. Although the conditional expectation is the optimal choice of the projection operator [6], in practice, it is challenging to estimate the conditional expectations with a finite set of data, especially for high-dimensional systems that are not fully resolved.

To address the aforementioned difficulties, we propose two novel ideas that enable data-driven learning for the MZ operators. First, we propose a concept that *any regression analysis* can be treated as *the* projection operator in the MZ formalism. Regression analysis is a statistical procedure to estimate the relationship between the dependent and independent variables. Such a relationship is determined by minimization of a prescribed error (risk) function, quantifying the discrepancy between model prediction and data. In our context of dynamical systems, the dependent variables would be those variables that we wish to predict in the future, and the independent variables would be those variables that we have already observed in the past. Second, we identify that the generalized fluctuation-dissipation (GFD) relationship, a self-consistent condition that relates the MZ memory kernels and the error of the memory-dependent dynamics, *should be* used to learn the memory kernels recursively. Building upon this concept, we have developed a systematic approach for learning MZ memory kernels, using snapshots of resolved variables. The resulting procedure is surprisingly simple, principled, and intuitive, despite a complicated operator-algebraic derivation. It is important to note that the memory kernels depend on the choice of the projection operator in the MZ formalism. As such, our procedure delivers memory kernels that are specific to the choice of regression model. We will demonstrate that with a linear regression model, the learned memory kernels converge nicely to the outcome of our recently proposed data-driven learning for Mori's projector [24]. Moreover, we will elaborate on how seemingly similar regression models can be associated with distinct projection operators, and how differentiating such a nuance could pave the way to enable learning the MZ memory kernels for reduced-order models with nonlinear closure schemes.

Before the technical presentation, it may be beneficial to provide a high-level characterization that contrasts our proposition to other statistical and machine-learning models motivated by Mori and Zwanzig's memory-dependent formulation. By doing so, we hope to provide a broader perspective on our work and the contributions it makes to the field. The key idea to differentiate our work from other models lies in the fact that MZ formalism is memory-dependent, but not all memory-dependent formulation is MZ. In our opinion, the memory kernels in MZ formalism are special because they satisfy the self-consistent GFD. As such, it would be a logical fallacy to call *any* memory-dependent dynamics an MZ model without enforcing GFD in the model structure, or without checking the GFD after learning. Unfortunately, many existing models fell into this logical fallacy. For example, [28] directly jumped from the MZ formalism to the recurrent neural network with long short term memory (RNN-

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

LSTM), claiming that the MZ memory can be learned by black-box RNN-LSTM without enforcing or checking the GFD. The same gap exists in [15] between the mathematical analysis based on MZ formalism and the computational RNN-LSTM architecture. Although GFD was used as an argument to adopt finite-memory length truncation, it is not explicitly enforced or checked that the (trained) LSTM memory satisfies the GFD. With evidence presented by [28] and [15], we cannot rule out that the RNN-LSTM learns other memory-dependent dynamics, for example, time delay embedding with which one augments the current state by the past history. Relevant data-driven delay embedded models include [14], which uses kernel methods, and [23], which uses Wiener projection and that employs infinitely long past history. Both delay-embedding studies aim to *bypass* quantification of the MZ memory kernel, as they argued that optimal delay embedding would result in vanishing MZ memories. A memory-dependent time-series analysis method termed NARMAX (nonlinear autoregressive moving average model with exogenous inputs [7]) also argues that its memory-dependence formulation resembles the MZ formalism without establishing the GFD. However, it is later established in [23] that the memory-dependent formulation NARMAX is an approximation of the Wiener projection, which has zero MZ memory. In contrast to these proposed methods, which postulate a memory-dependence in the statistical learning model without enforcing the GFD, our proposed approach has GFD built in. In fact, GFD is so critical that it turned out to be the key equation to establish our proposed recursive procedure. To our best knowledge, such a procedure does not exist in the literature.

To support the claim that our proposition can be applied to a wide range of data-driven models parametrized by regression, we will present numerical experiments on various nonlinear regression models, including polynomial and ridge regression, fully connected neural networks (FCNN), and convolutional neural networks (CNN). It is not our intention to show that these simple regression models outperform existing data-driven and regression-based models. Rather, we would like to use these simple examples to illustrate the mechanism of the proposed procedure, the learned MZ (with enforced GFD) memories, and how including the memory contribution can improve the predictions. We reemphasize that our proposition is not limited to these simple regression models. With the identification of a regression analysis as the MZ projection operator, our proposed procedure can be applied to a wide array of data-driven models parametrized by regression (e.g., [5, 22, 27, 42, 51]) for quantifying their MZ memory kernels and for improving their predictions. Consequently, our proposition has the potential to significantly broaden the practical scope of the MZ formalism.

2. Background and terminology.

2.1. Full dynamical systems.

Following the notation in [24], we consider an autonomous and deterministic dynamical system in $\mathbb{R}^D$ following a continuous-time evolutionary equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{\phi}(t) = \mathbf{R}\left(\boldsymbol{\phi}\left(t\right)\right), \tag{2.1}$$

where $\boldsymbol{\phi}$ is the state in the phase space $\mathbb{R}^D$ and $\mathbf{R}: \mathbb{R}^D \to \mathbb{R}^D$ is the vector field of the system. We will assume that the system has a unique $\boldsymbol{\phi}(t)$ $\forall t \geq 0$. We will denote the solution of (2.1) with the initial condition $\boldsymbol{\phi}_0 \in \mathbb{R}^D$ by $\boldsymbol{\phi}(t; \boldsymbol{\phi}_0)$. We will use the standard notation and denote scalars by symbols in the normal font and vectors by symbols in bold.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

2.2. Observables. In reduced-order modeling, we aim to prescribe the evolutionary equations for $M < D$ real-valued functions of the state variable $\boldsymbol{\phi}$, $g_i : \mathbb{R}^D \to \mathbb{R}$, $i = 1 \dots M$. We exclusively consider real-valued functions, but the theory can be straightforwardly generalized to complex-valued functions. We refer to these functions as the observables. In other contexts, such as material science or statistical mechanics, observables can also be referred to as descriptors, coarse-grained variables, or dynamic variables. We refer to an "observation" at time t as applying the set of observables to the system's state $\boldsymbol{\phi}(t; \boldsymbol{\phi}_0)$, that is, a real-valued array $\{g_i(\boldsymbol{\phi}(t; \boldsymbol{\phi}_0))\}_{i=1}^M$. We remark that one can define an observable π_i to extract the ith component of the state $\boldsymbol{\phi}$ at time t (that is, $\pi_i(\boldsymbol{\phi}(t; \boldsymbol{\phi}_0)) = \boldsymbol{\phi}_i(t; \boldsymbol{\phi}_0)$ [23]). In general, observables g must be square integrable functions of the full state $\boldsymbol{\phi}$ against some distribution; see section 2.5. We will use $\mathbf{g}$ (resp., $\mathbf{g}(\boldsymbol{\phi})$) to denote a vectorized observable $[g_1, g_2, \dots g_M]^T$ (resp., observations, $[g_1(\boldsymbol{\phi}), g_2(\boldsymbol{\phi}), \dots g_M(\boldsymbol{\phi})]^T$).

2.3. Discrete-time snapshot data. Throughout this study, we assume that the full system has been simulated and observed at discrete times, and the observations form a data set for learning the MZ operators. Specifically, we rely on samples drawn from a chosen initial distribution μ (see section 2.5) to collect the statistics of the unresolved information. The full system is prepared at $N \gg 1$ independently and identically sampled $\boldsymbol{\phi}_0^{[i]} \sim \mu$, $i = 1 \dots N$. The system is then simulated and the values of the observations are registered at uniformly distributed discrete times $k\Delta$, $k = 0 \dots K-1$. We refer to these observations as the *snapshots*. Without loss of generality, we assume $\Delta = 1$ by choosing an appropriate unit of time, unless it is specified otherwise. The full discrete-time observations thus form an $N \times M \times K$ data matrix $\mathbf{D}$ whose (i, j, k)-entry is $g_j(\boldsymbol{\phi}(k-1; \boldsymbol{\phi}_0^{[i]}))$. For those systems which have natural or physical invariant measure, it is desirable to bypass sampling from a prespecified initial distribution and use the long-time statistics instead. In this case, we generate a single but long trajectory from a randomly selected initial condition $\boldsymbol{\phi}_0^{\mathrm{r}}$. After the initial transient time t_c, we make $K + N$ observations for every $\Delta = 1$. We use these $K + N$ observations to inform the statistics of the invariant measure. The assumptions underlying the sufficiency of using $K + N$ observations are described below. The data matrix $\mathbf{D}$ in this case is defined as an $N \times M \times K$ matrix whose (i, j, k)-entry is $g_j(\boldsymbol{\phi}(t_c + i + k - 1; \boldsymbol{\phi}_0^{\mathrm{r}}))$.

2.4. Discrete-time dynamical systems. To match to the discrete-time snapshot data, we rewrite the evolutionary equation (2.1) into a discrete-time form:

$$\boldsymbol{\phi}(t+1) = \mathbf{F}(\boldsymbol{\phi}(t)), \tag{2.2}$$

where $\mathbf{F} : \mathbb{R}^D \to \mathbb{R}^D$ is the flow map. We assume that the map exists and is unique for any finite time $t \in \mathbb{Z}_{\geq 0}$. Our formulation is readily applicable to those generically discrete-time dynamical systems, in which we begin with the discrete-time formulation (2.2) instead of (2.1).

2.5. A chosen distribution and square integrable functions. We assume that the number of samples N is sufficiently large to capture the statistics of the time-dependent distribution induced by the deterministic dynamics from the initial distribution μ, i.e., the solution of the generalized Liouville equation [13] corresponding to (2.1):

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

$$\frac{\partial}{\partial t}\rho(\boldsymbol{\phi},t) = -\sum_{i=1}^{N}\frac{\partial}{\partial \boldsymbol{\phi}_i}\left[\mathbf{R}_i(\boldsymbol{\phi})\,\rho(\boldsymbol{\phi},t)\right] \text{ with the initial data } \rho(\boldsymbol{\phi},0) = \mu(\boldsymbol{\phi}), \tag{2.3}$$

where $\mu(\boldsymbol{\phi})$ is the probability density function of the measure μ, assuming the measure μ is dominated by the Lebesgue measure in the state space $\mathbb{R}^D$ so $\mu(\mathrm{d}\boldsymbol{\phi}) = \mu(\boldsymbol{\phi})\,\mathrm{d}\boldsymbol{\phi}$. When using the single trajectory to sample the ergodic distribution, we assume that N is sufficiently large such that the collected $N+K$ samples faithfully capture the statistics of the ergodic measure, i.e., the stationary solution ρ_* of (2.3) satisfying

$$0 = \sum_{i=1}^{N}\frac{\partial}{\partial \boldsymbol{\phi}_i}\left[\mathbf{R}_i(\boldsymbol{\phi})\,\rho_*(\boldsymbol{\phi})\right]. \tag{2.4}$$

In both cases, we assume that each of the observables, g_i, is a square integrable function against the induced measure. That is, if one chooses to learn from a time-dependent distribution (equation (2.3)), $\int_{\mathbb{R}^D}\rho(\boldsymbol{\phi},t)\,g_i^2(\boldsymbol{\phi})\,\mathrm{d}\boldsymbol{\phi} < \infty$ for $0 \le t \le (K-1)\,\Delta$, or if one chooses to learn from a stationary distribution (equation (2.4)), $\int_{\mathbb{R}^D}\rho_*(\boldsymbol{\phi})\,g_i^2(\boldsymbol{\phi})\,\mathrm{d}\boldsymbol{\phi} < \infty$.

2.6. The projection operator. Suppose all the M resolved observables are independent; we shall need another $D-M$ independent unresolved observables to uniquely specify the system's state. Denote these unresolved observables, which are also real-valued functions of the system's state $\boldsymbol{\phi}$, by $u_j : \mathbf{R}^D \to \mathbb{R}$, $j = 1 \ldots D-M$. Importantly, observations of these unresolved variables, $u_j(\boldsymbol{\phi})$, are not accessible. Consider a real-valued function of the full state expressed in both the resolved and unresolved observations, $h(\{g_i(\boldsymbol{\phi})\}_{i=1}^{M}, \{u_j(\boldsymbol{\phi})\}_{j=1}^{D-M})$. A projection operator $\mathcal{P}$ is an operator mapping h to another real-valued function $(\mathcal{P}h) : \mathbb{R}^D \to \mathbb{R}$ which only depends on the resolved observations. MZ formalism utilizes this projection operator and decomposes any function $h(\{g_i(\boldsymbol{\phi})\}_{i=1}^{M}, \{u_j(\boldsymbol{\phi})\}_{j=1}^{D-M})$ into $(\mathcal{P}h)(\{g_i(\boldsymbol{\phi})\}_{i=1}^{M}) + h_\perp(\{g_i(\boldsymbol{\phi})\}_{i=1}^{M}, \{u_j(\boldsymbol{\phi})\}_{j=1}^{D-M})$, where $h_\perp := h - \mathcal{P}h$. Note that the domain of $(\mathcal{P}h)$ is still both the resolved and unresolved observations, except that the function does not explicitly depend on the unresolved $u_j(\boldsymbol{\phi})$. A projection operator $\mathcal{P}$ must satisfy the property $\mathcal{P}^2 = \mathcal{P}$, so that for any h, $\mathcal{P}^n h = \mathcal{P}h$, $n \in \mathbb{N}$. Such a property is critical in the derivation of the MZ formalism [55].

2.7. Discrete-time Mori–Zwanzig formalism. Following the derivations in [8, 23, 24, 49], the discrete-time MZ formalism prescribes the exact evolutionary equations of the observations, that for any initial condition $\boldsymbol{\phi}_0 \in \mathbb{R}^D$,

$$\mathbf{g}_{n+1}(\boldsymbol{\phi}_0) = \boldsymbol{\Omega}^{(0)}(\mathbf{g}_n(\boldsymbol{\phi}_0)) + \sum_{\ell=1}^{n}\boldsymbol{\Omega}^{(\ell)}(\mathbf{g}_{n-\ell}(\boldsymbol{\phi}_0)) + \mathbf{W}_n(\boldsymbol{\phi}_0). \tag{2.5}$$

Here, $\mathbf{g}_n : \mathbb{R}^D \to \mathbb{R}^M$ is an $M \times 1$ vector function of the initial state $\boldsymbol{\phi}_0$ such that

$$\mathbf{g}_n(\boldsymbol{\phi}_0) \triangleq \mathbf{g}(\boldsymbol{\phi}(n;\boldsymbol{\phi}_0)) \equiv \mathbf{g}(\mathbf{F}^n(\boldsymbol{\phi}_0)), \tag{2.6}$$

noting that $\mathbf{g}_0 \equiv \mathbf{g}$. In other words, denote the finite-time (Δ) Koopman operator [20] by $\mathcal{K}$, $\mathbf{g}_n := \mathcal{K}^n\mathbf{g}$. For clarity, we write each component of this vector function as $\mathbf{g}_{n,i} := \mathcal{K}^n g_i$ if

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

needed. Note that $\mathbf{g}_{0,i} = \mathbf{g}_i$. Equation (2.5), referred to as the GLE, is the central result of the MZ formalism. It states that the vectorized observation at time $n+1$, $\mathbf{g}(\boldsymbol{\phi}(n+1;\boldsymbol{\phi}_0))$, can be decomposed into three parts: (1) a Markovian function $\boldsymbol{\Omega}^{(0)} : \mathbb{R}^M \to \mathbb{R}^M$ mapping the observations at time n to an $M \times 1$ vector, (2) a series of functions $\boldsymbol{\Omega}^{(\ell)} : \mathbb{R}^M \to \mathbb{R}^M$, $\ell = 1, 2, \ldots$, mapping past observations with a lag ℓ to $M \times 1$ vectors, and (3) the *orthogonal dynamics* that consist of observables $\mathbf{W}_n : \mathbb{R}^D \to \mathbb{R}^M$, $n = 0, 1 \ldots$, mapping any initial state $\boldsymbol{\phi}_0$ to an $M \times 1$ vector. Interested reader can find a detailed discussion and physical interpretation of the GLE (2.5) in [24].

Because the GLE (2.5) holds true for every initial condition $\boldsymbol{\phi}_0$, we can drop the function evaluation at $\boldsymbol{\phi}_0$ part and write succinctly an equation for the observables

$$\mathbf{g}_{n+1} = \sum_{\ell=0}^{n} \boldsymbol{\Omega}^{(\ell)} \mathbf{g}_{n-\ell} + \mathbf{W}_n. \tag{2.7}$$

In writing so, we overload $\boldsymbol{\Omega}^{(\ell)}$, which are functions in (2.5), to denote operators in (2.7). That is, in (2.7), $\boldsymbol{\Omega}^{(\ell)}$ are functional operators mapping an $M \times 1$ vector function, each of whose component is an $L^2(\mathbb{R}^D, \mu)$-function, to another $M \times 1$ vector function, each of whose component is an $L^2(\mathbb{R}^D, \mu)$-function again. We thus refer to $\boldsymbol{\Omega}^{(\ell)}$ as the "MZ operators." We will adopt the slightly abused notation that the same symbol $\boldsymbol{\Omega}^{(\ell)}$ is used to denote both the functions in equations of observations (such as (2.5)) and operators in equations of observables (such as (2.7)).

The operators $\boldsymbol{\Omega}^{(\ell)}$ and $\mathbf{W}_n$ depend on the choice of the projection operator $\mathcal{P}$, the choice of the vectorized observable $\mathbf{g}$, and the finite-time (Δ) Koopman operator $\mathcal{K}$ [8, 23]:

$$\boldsymbol{\Omega}^{(n)} := \mathcal{P}\mathcal{K}\left[(1-\mathcal{P})\mathcal{K}\right]^n, \tag{2.8}$$

$$\mathbf{W}_n := \left[(1-\mathcal{P})\mathcal{K}\right]^{n+1} \mathbf{g} \tag{2.9}$$

with $n \in \mathbb{Z}_{\geq 0}$. Because $\mathcal{P}\mathbf{W}_n = 0$ (from the fact $\mathcal{P}^2 = \mathcal{P}$), $\mathbf{W}_n$ is often referred to as the orthogonal dynamics. From these definitions, it is straightforward to identify the following equation, which relates the memory operators to the orthogonal dynamics:

$$\boldsymbol{\Omega}^{(n)} \mathbf{g} \equiv \mathcal{P}\mathcal{K}\mathbf{W}_{n-1} \forall n \in \mathbb{N}. \tag{2.10}$$

The meaning of the above equation is more transparent when we apply the operators to any state $\boldsymbol{\phi}_0 \in \mathbb{R}^D$,

$$\boldsymbol{\Omega}^{(n)}(\mathbf{g}(\boldsymbol{\phi}_0)) \equiv \mathcal{P}\mathcal{K}\mathbf{W}_{n-1}(\boldsymbol{\phi}_0) \equiv \mathcal{P}((\mathbf{W}_{n-1} \circ \mathbf{F})(\boldsymbol{\phi}_0)) \forall n \in \mathbb{N}, \tag{2.11}$$

in which we applied the definition of the Koopman operator $\mathcal{K}$ and used $\circ$ to denote function composition. The above equation is the GFD relation. Below, we will demonstrate that the GFD enables data-driven learning of the operators $\boldsymbol{\Omega}^{(\ell)}$'s.

2.8. Regression. Here, we provide a short description of the regression analysis. In regression analysis, the goal is to identify an optimal model f as a function of the independent variables $\mathbf{x}$ for predicting the dependent variables $\mathbf{y}$. We first focus on a scalar variable y, which can be one of the many components of a vector $\mathbf{y}$. The model f is defined as a family

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

of functions $f(\mathbf{x};\boldsymbol{\theta})$ of the independent variables $\mathbf{x}$, parametrized by an array of parameters $\boldsymbol{\theta}$. The assumption of the (homoscedastic) regression model is that, in the most general form, there exists a random variable ε such that the dependent variable Y can be expressed as

$$Y(\mathbf{x}) = f(\mathbf{x};\boldsymbol{\theta}_*) + \varepsilon. \tag{2.12}$$

Here, we use the standard capital notation to denote a random variable Y. Given a list of paired realizations $\{\mathbf{x}^{[i]}, y^{[i]}\}_{i=1}^N$, the best-fit parameter $\boldsymbol{\theta}_*$ minimizes the differences between the model prediction $f(\mathbf{x};\boldsymbol{\theta})$ and the dependent variable Y. The best-fit parameter $\boldsymbol{\theta}_*$ is solved by minimizing a negative log-likelihood function that depends on the noise distribution. For example, for normally distributed ε, the negative log-likelihood function is proportional to the mean squared error (MSE), so the best-fit parameter is the minimizer

$$\boldsymbol{\theta}_* := \arg\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{i=1}^{N}\left(y^{[i]} - f\left(\mathbf{x}^{[i]};\boldsymbol{\theta}\right)\right)^2. \tag{2.13}$$

When the dependent variables are vectors, that is, $\mathbf{y}$ is $P > 1$ dimensional, the cost function can be chosen as the sum of the MSE of M independent regressions:

$$\boldsymbol{\theta}_* := \arg\min_{\boldsymbol{\theta}} \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{P}\left(y_j^{[i]} - f_j\left(\mathbf{x}^{[i]};\boldsymbol{\theta}\right)\right)^2. \tag{2.14}$$

The above equation comes with three assumptions about the noise model. First, in each component j of the P-dimensional $\mathbf{y}$, the noise ε_j is normally distributed. Second, the noise ε_j is independently and identically distributed in each of the dimensions. Third, the noise is homoscedastic, which means that the variance does not depend on the independent variable $\mathbf{x}$. We remark that with a different noise model (e.g., correlated or heteroscedastic noise), the cost function can take a different form. Nevertheless, our method does not depend on the specific form of the cost function, i.e., it is not necessary to use MSE. Thus, in the analysis below, we express a general cost function by

$$C\left(\boldsymbol{\theta};\left\{\mathbf{x}^{[i]},\mathbf{y}^{[i]}\right\}_{i=1}^{N}\right), \tag{2.15}$$

so the best-fit parameters $\boldsymbol{\theta}_*$ are obtained by solving a general optimization problem:

$$\boldsymbol{\theta}_* := \arg\min_{\boldsymbol{\theta}} C\left(\boldsymbol{\theta};\left\{\mathbf{x}^{[i]},\mathbf{y}^{[i]}\right\}_{i=1}^{N}\right). \tag{2.16}$$

2.9. Regression-based projection operators. Using the notation defined above, we treat the observations at the time $s \geq 0$ as the independent variables and the evaluations of any real-valued function $h : \mathbb{R}^D \to \mathbb{R}$ of the full state $\boldsymbol{\phi}(t;\boldsymbol{\phi}_0)$ at a later time $t > s$ as the dependent variable. The samples of the dependent and the independent variables are generated from the fully resolved simulation, whose initial condition $\boldsymbol{\phi}_0^{[i]}$ is sampled from the distribution μ. Note that the time-dependent full states $\boldsymbol{\phi}(s;\boldsymbol{\phi}_0^{[i]})$ and $\boldsymbol{\phi}(t;\boldsymbol{\phi}_0^{[i]})$ are neither resolved nor accessed, but the observables $\mathbf{g}_s(\boldsymbol{\phi}_0^{[i]}) := \mathbf{g}(\boldsymbol{\phi}(s;\boldsymbol{\phi}_0^{[i]}))$ and $h_t(\boldsymbol{\phi}_0^{[i]}) := h(\boldsymbol{\phi}(t;\boldsymbol{\phi}_0^{[i]}))$ are measured and

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

registered as we simulate the dynamics. We use the samples to regress $h_t(\boldsymbol{\phi}_0^{[i]})$ on $\mathbf{g}_s(\boldsymbol{\phi}_0^{[i]})$ using a family of functions $f(\mathbf{g}_s(\boldsymbol{\phi}_0^{[i]});\boldsymbol{\theta})$ by minimizing a chosen cost function:

$$\boldsymbol{\theta}_* := \arg\min_{\theta} C\left(\boldsymbol{\theta}; \left\{\mathbf{g}_s\left(\boldsymbol{\phi}_0^{[i]}\right), h_t\left(\boldsymbol{\phi}_0^{[i]}\right)\right\}_{i=1}^{N}\right). \tag{2.17}$$

The best fit $f(\cdot;\boldsymbol{\theta}_*) : \mathbb{R}^M \to \mathbb{R}$ is now a function that depends on only the resolved initial observations at time s, $\mathbf{g}_s(\boldsymbol{\phi}_0)$. It is clear that another regression of $f(\mathbf{g}_s(\boldsymbol{\phi}_0);\boldsymbol{\theta}_*)$ on the independent variables $\mathbf{g}_s(\boldsymbol{\phi}_0)$ results in $f(\cdot;\boldsymbol{\theta}_*)$ again. As such, regression is a projection of h_t to $f(\mathbf{g}_s(\boldsymbol{\phi}_0);\boldsymbol{\theta}_*)$:

$$(\mathcal{P}h_t)(\mathbf{g}_s(\boldsymbol{\phi}_0)) = f(\mathbf{g}_s(\boldsymbol{\phi}_0);\boldsymbol{\theta}_*). \tag{2.18}$$

In summary, with the choice of using regression for projection, applying $\mathcal{P}$ to any function h_t of the full-system state $\boldsymbol{\phi}$ results in the best-fit parametric function $f(\cdot;\boldsymbol{\theta}_*)$ of the reduced-order observations $\mathbf{g}_s(\boldsymbol{\phi}_0)$. With a cost function such as (2.14), a straightforward generalization can be made to those vector functions $\mathbf{h}_t : \mathbb{R}^D \to \mathbb{R}^P$ that $(\mathcal{P}\mathbf{h}_t)(\cdot) = \mathbf{f}(\cdot;\boldsymbol{\theta}_*)$, where $\mathbf{f}$ is a parametric family of $\mathbb{R}^M \to \mathbb{R}^P$ functions. We remark that the regression-based projection operator may no longer be the projection operators commonly adopted in existing MZ models, e.g., Mori's, finite-rank, and Zwanzig's projection operators. In addition, the projection operator defined by the regression analysis may no longer be an orthogonal projection, with respect to the typical inner product defined in L^2-spaces.

3. Theoretical results.

3.1. Learning Mori–Zwanzig operators with regression-based projection using snapshot data.

We now use the snapshot data matrix $\mathbf{D}$ to learn the MZ operators $\boldsymbol{\Omega}^{(n)}$ and orthogonal dynamics $\mathbf{W}_n$, $n \in \mathbb{Z}_{\geq 0}$. As will be seen below, this can be achieved by iteratively performing a regression problem as a projection, quantifying the residuals as the orthogonal dynamics, and utilizing the GFD to define the next-order regression problem. We begin with setting $n = 0$; the GLE (2.5) with the expression (2.8) states

$$\mathbf{g}_1(\boldsymbol{\phi}_0) = \mathcal{P}\mathcal{K}(\mathbf{g}(\boldsymbol{\phi}_0)) + \mathbf{W}_0(\boldsymbol{\phi}_0) = \mathcal{P}[\mathbf{g}_1(\boldsymbol{\phi}_0)] + \mathbf{W}_0(\boldsymbol{\phi}_0) \tag{3.1}$$

for any initial state $\boldsymbol{\phi}_0$. We used the definition of the Koopman operator $\mathcal{K}(\mathbf{g}(\boldsymbol{\phi}_0)) = \mathbf{g}_1(\boldsymbol{\phi}_0) := \mathbf{g}(\boldsymbol{\phi}(1;\boldsymbol{\phi}_0))$, which are the observations made at one time step ahead of the initial condition $\boldsymbol{\phi}_0$. These observations can be accessed in the snapshot data matrix: they are simply $\mathbf{D}(\cdot,\cdot,2)$. So can the $\mathbf{g}(\boldsymbol{\phi}_0)$, which are $\mathbf{D}(\cdot,\cdot,1)$. The regression-based $\mathcal{P}$ now projects $\mathbf{g}_1$ to an optimal function of $\mathbf{g}(\boldsymbol{\phi}_0)$, by regressing $\mathbf{D}(\cdot,\cdot,2)$ on $\mathbf{D}(\cdot,\cdot,1)$. For example, using the MSE cost function (2.14), we solve the following optimization problem for projection:

$$C^{(0)}(\boldsymbol{\theta};\mathbf{D}) := \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M}\left[\mathbf{D}(i,j,2) - \mathbf{f}_j\left(\{\mathbf{D}(i,k,1)\}_{k=1}^{M};\boldsymbol{\theta}\right)\right]^2, \tag{3.2a}$$

$$\boldsymbol{\theta}_*^{(0)} := \arg\min_{\theta} C^{(0)}(\boldsymbol{\theta};\mathbf{D}), \tag{3.2b}$$

$$\boldsymbol{\Omega}^{(0)}(\cdot) = \mathbf{f}\left(\cdot;\boldsymbol{\theta}_*^{(0)}\right), \tag{3.2c}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

where $\mathbf{f}(\cdot;\boldsymbol{\theta}):\mathbb{R}^M \to \mathbb{R}^M$ is a family of regressional functions defined by the user. Here, the superscript (0) is used to denote the cost function ($C^{(0)}$) and the solution of the parameters ($\boldsymbol{\theta}^{(0)}$) of the 0th-order regression analysis. Once the best-fit model is solved, the orthogonal dynamics $\mathbf{W}_0$ is identified as the *residual* of the regression problem from (3.1). That is, for the ith sample initial condition $\boldsymbol{\phi}_0^{[i]}$, the jth component of the orthogonal dynamics at time 0 is

$$\begin{aligned}(\mathbf{W}_0)_j\left(\boldsymbol{\phi}_0^{[i]}\right) &= \left[\mathbf{g}_1\left(\boldsymbol{\phi}_0^{[i]}\right) - \mathcal{P}\left(\mathbf{g}_1\left(\boldsymbol{\phi}_0^{[i]}\right)\right)\right]_j \\ &= \mathbf{D}(i,j,2) - \mathbf{f}_j\left(\{\mathbf{D}(i,k,1)\}_{k=1}^M;\boldsymbol{\theta}_*^{(0)}\right). \end{aligned} \tag{3.3}$$

We now use mathematical induction to show that we can extract $\boldsymbol{\Omega}^{(n+1)}$ and samples of $\mathbf{W}_{n+1}$, given $\boldsymbol{\Omega}^{(\ell)}$, $\ell = 0\ldots n$, and the snapshot data matrix $\mathbf{D}$. We first use GLE (2.7) to express $\mathbf{W}_n$ in terms of $\boldsymbol{\Omega}^{(\ell)}$ and $\mathbf{g}_{n-\ell}$:

$$\mathbf{W}_n = \mathbf{g}_{n+1} - \sum_{\ell=0}^{n} \boldsymbol{\Omega}^{(\ell)}(\mathbf{g}_{n-\ell}). \tag{3.4}$$

Then, we use the GFD, (2.11), to construct $\boldsymbol{\Omega}^{(n+1)}$:

$$\begin{aligned}\boldsymbol{\Omega}^{(n+1)}(\mathbf{g}) &= \mathcal{P}\mathcal{K}\mathbf{W}_n = \mathcal{P}\left[\mathcal{K}\mathbf{g}_{n+1} - \mathcal{K}\sum_{\ell=0}^{n}\boldsymbol{\Omega}^{(\ell)}(\mathbf{g}_{n-\ell})\right] \\ &= \mathcal{P}\left(\mathbf{g}_{n+2} - \sum_{\ell=0}^{n}\boldsymbol{\Omega}^{(\ell)}(\mathbf{g}_{n-\ell+1})\right). \end{aligned} \tag{3.5}$$

We evaluate the above function on an initial state $\boldsymbol{\phi}_0$ and use the definition (2.6) to obtain

$$\boldsymbol{\Omega}^{(n+1)}(\mathbf{g}(\boldsymbol{\phi}_0)) = \mathcal{P}\left(\mathbf{g}(\boldsymbol{\phi}(n+2;\boldsymbol{\phi}_0)) - \sum_{\ell=0}^{n}\boldsymbol{\Omega}^{(\ell)}(\mathbf{g}(\boldsymbol{\phi}(n-\ell+1;\boldsymbol{\phi}_0)))\right). \tag{3.6}$$

In the right-hand side of this expression, $\boldsymbol{\Omega}^{(\ell)}$, $\ell = 0\ldots n$, is the given best-fit parametric functions $\mathbf{f}(\cdot;\boldsymbol{\theta}_*^{(\ell)})$. Note that for a particular initial sample $\boldsymbol{\phi}^{[i]}$, all the observations in the above equation are registered in the data matrix $\mathbf{D}$.

$$\mathbf{g}\left(\boldsymbol{\phi}_0^{[i]}\right) = \mathbf{D}(i,\cdot,1), \tag{3.7a}$$

$$\mathbf{g}\left(\boldsymbol{\phi}\left(n+2;\boldsymbol{\phi}_0^{[i]}\right)\right) = \mathbf{D}(i,\cdot,n+3), \tag{3.7b}$$

$$\mathbf{g}\left(\boldsymbol{\phi}\left(n-\ell+1;\boldsymbol{\phi}_0^{[i]}\right)\right) = \mathbf{D}(i,\cdot,n-\ell+2). \tag{3.7c}$$

Similar to (3.2), we now apply the projection operator $\mathcal{P}$, which is a regression on the corresponding variables:

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

$$\mathbf{y}_{j,n+1}^{[i]} := \mathbf{D}(i,j,n+3) - \sum_{\ell=0}^{n} \mathbf{f}_j\left(\{\mathbf{D}(i,k,n-\ell+2)\}_{k=1}^{M}; \boldsymbol{\theta}_*^{(\ell)}\right), \tag{3.8a}$$

$$C^{(n+1)}(\boldsymbol{\theta};\mathbf{D}) := \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M}\left[\mathbf{y}_{j,n+1}^{[i]} - \mathbf{f}_j\left(\{\mathbf{D}(i,k,1)\}_{k=1}^{M};\boldsymbol{\theta}\right)\right]^2, \tag{3.8b}$$

$$\boldsymbol{\theta}_*^{(n+1)} := \arg\min_{\boldsymbol{\theta}} C^{(n+1)}(\boldsymbol{\theta};\mathbf{D}), \tag{3.8c}$$

$$\boldsymbol{\Omega}^{(n+1)} \equiv \mathbf{f}\left(\cdot;\boldsymbol{\theta}_*^{(n+1)}\right). \tag{3.8d}$$

Again, the superscript $(n+1)$ here denotes the cost function ($C^{(n+1)}$), best-fit parameters $\boldsymbol{\theta}_*^{(n+1)}$, and best-fit parametric functions $\mathbf{f}(\cdot;\boldsymbol{\theta}_*^{(n+1)})$ of this $(n+1)$th-order regression analysis. By induction, the above procedure iteratively extracts $\boldsymbol{\Omega}^{(n)}(\cdot) = \mathbf{f}(\cdot;\boldsymbol{\theta}_*^{(n)})$, $n = 0 \ldots K-2$ (in total, $K-1$ operators), using the snapshot data matrix $\mathbf{D}$. Note that the family of regressional functions $\mathbf{f}$ is kept the same for all ℓ to ensure a consistent projection operator $\mathcal{P}$. Notably, both the orthogonality $\mathcal{P}\boldsymbol{W}_n = 0$ and the GFD $\boldsymbol{\Omega}^{(n+1)}(\mathbf{g}) = \mathcal{P}\mathcal{K}\mathbf{W}_n$ are built in by construction. We summarize the procedure in Algorithm 3.1.

The above procedure iteratively learns the operators $\boldsymbol{\Omega}^{(\ell)}$ in the GLE (2.7). The GLE is a particular form of memory-dependent dynamics, which is specific to the choices of the observables $\mathbf{g}$ and projection operator $\mathcal{P}$. There are other forms of memory-dependent dynamics

Algorithm 3.1 Extracting the regression-based MZ operators. **Input:** A set of observables $\mathbf{g}: \mathbb{R}^D \to \mathbb{R}^M$; the data matrix $\mathbf{D}$ of the snapshots of the observations (see section 2.3); a family of regressional functions $\mathbf{f}(\cdot;): \mathbb{R}^M \to \mathbb{R}^M$ parametrized by fitting parameters $\boldsymbol{\theta}$; the cost function of the regression analysis, $C(\boldsymbol{\theta};\{\mathbf{y}^{[i]},\mathbf{x}^{[i]}\}_{i=1}^N)$ (see section 2.8). **Output:** The Markov operator $\boldsymbol{\Omega}^{(0)}: \mathbb{R}^M \to \mathbb{R}^M$; memory operators $\boldsymbol{\Omega}^{(n)}: \mathbb{R}^M \to \mathbb{R}^M$, $n \in \{1,2,\ldots K-2\}$; orthogonal dynamics $\mathbf{W}_n$, $n \in \{1,2,\ldots K-3\}$.

$\mathbf{x}_j^{[i]} \leftarrow \mathbf{D}(i,j,1)$
$\mathbf{y}_{j,0}^{[i]} \leftarrow \mathbf{D}(i,j,2)$
$\boldsymbol{\theta}_*^{(0)} \leftarrow \arg\min_{\boldsymbol{\theta}} C\left(\boldsymbol{\theta};\left\{\mathbf{y}_0^{[i]},\mathbf{x}^{[i]}\right\}_{i=1}^N\right)$
$\boldsymbol{\Omega}^{(0)}(\cdot) \leftarrow \mathbf{f}\left(\cdot;\boldsymbol{\theta}_*^{(0)}\right)$
for n in $\{1,\ldots K-2\}$ **do**
  $\mathbf{y}_{j,n}^{[i]} \leftarrow \mathbf{D}(i,j,n+2) - \sum_{\ell=0}^{n-1}\mathbf{f}_j\left(\{\mathbf{D}(i,k,n-\ell+1)\}_{k=1}^M;\boldsymbol{\theta}_*^{(\ell)}\right)$
  $\mathbf{W}_{n-1}^{[i]} \leftarrow \mathbf{y}_n^{[i]}$
  $\boldsymbol{\theta}_*^{(n)} \leftarrow \arg\min_{\boldsymbol{\theta}} C\left(\boldsymbol{\theta};\left\{\mathbf{y}_n^{[i]},\mathbf{x}^{[i]}\right\}_{i=1}^N\right)$
  $\boldsymbol{\Omega}^{(n)}(\cdot) \leftarrow \mathbf{f}\left(\cdot;\boldsymbol{\theta}_*^{(n)}\right)$
end for
Output $\boldsymbol{\Omega}^{(n)}$ and $\mathbf{W}_n$

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

which do not rely on an a priori selected projection operator, for example, the Hankel-DMD method [3]. Numerical comparison of these methods and a detailed discussion will be provided below in sections 4.2 and 5.

3.2. Making predictions and modeling orthogonal dynamics. Our goal is to iteratively use GLE (2.5) to make multistep predictions for the resolved observables, once the operators $\mathbf{\Omega}^{(\ell)}$'s are learned. For prediction, it is convenient to shift the time index and write the "present time" as $t=0$, assuming the system initiated from $\boldsymbol{\phi}_{-T}$ at $t=-T$ in the past ($T \in \mathbb{Z}_{\geq 0}$). We assume that the present and a finite number of past times, $\mathbf{g}_{-\ell}\left(\boldsymbol{\phi}_{-T}\right)$, $\ell = 0, 1, \dots H-1 \leq T$, are given. We will make predictions for the resolved variables at a future horizon $t=k \geq 1$, $\mathbf{g}_k\left(\boldsymbol{\phi}_{-T}\right)$. To achieve this, we first use a truncated GLE to predict $\mathbf{g}_1$:

$$\mathbf{g}_1\left(\boldsymbol{\phi}_{-T}\right) = \sum_{\ell=0}^{H-1} \mathbf{\Omega}^{(\ell)}\left(\mathbf{g}_{-\ell}\left(\boldsymbol{\phi}_{-T}\right)\right) + \mathbf{W}_T\left(\boldsymbol{\phi}_{-T}\right). \tag{3.9}$$

This is a truncated GLE because only a finite length of the history is provided; terms $\mathbf{\Omega}^{(\ell)}\left(\mathbf{g}_{-\ell}\left(\boldsymbol{\phi}_{-T}\right)\right)$, $\ell = H \dots T$, are neglected. We show below with numerical evidence that this truncation is justifiable. Once $\mathbf{g}_1\left(\boldsymbol{\phi}_{-T}\right)$ is obtained, we accumulate the predicted $\mathbf{g}_1\left(\boldsymbol{\phi}_{-T}\right)$ into the series of observations, $\mathbf{g}_{-\ell}\left(\boldsymbol{\phi}_{-T}\right)$, $\ell = -1, 0, 1 \dots H-1$, and use this accumulated series to make prediction of $\mathbf{g}_2\left(\boldsymbol{\phi}_{-T}\right)$ using the truncated GLE (3.9) again:

$$\mathbf{g}_2\left(\boldsymbol{\phi}_{-T}\right) = \sum_{\ell=0}^{H-1} \mathbf{\Omega}^{(\ell)}\left(\mathbf{g}_{1-\ell}\left(\boldsymbol{\phi}_{-T}\right)\right) + \mathbf{W}_{T+1}\left(\boldsymbol{\phi}_{-T}\right). \tag{3.10}$$

The procedure continues until the desired horizon $m \geq 1$ is reached. Note that this procedure only needs the first H operators $\mathbf{\Omega}^{(\ell)}$, $\ell = 0 \dots H-1$. Clearly, the orthogonal dynamics $\mathbf{W}_n$, $n = T \dots T+m-1$, are needed in the procedure. Recall that the orthogonal dynamics encode the information in the unresolved space (equation (2.9)), so they cannot be obtained by data-driven approaches. We must then rely on modeling. However, modeling the orthogonal dynamics is challenging: it is equivalent to modeling all the $D-M$ unresolved degrees of freedom. In addition, the modeling is specific to the full dynamical system, the choice of the resolved observables, and the choice of the projection operator.

A best-case scenario is that the resolved observables are reasonably selected, and an expressive enough regression model is chosen. With a sufficiently long history length H, this leads to negligible orthogonal dynamics, i.e., the residuals of the regression analysis are small enough to be negligible. In the numerical experiments below, we assume this best-case scenario and proceed with a trivial zero-noise model $\mathbf{W}_n = 0$, $n \geq T$, although finite residuals are observed in the experiments. Prediction with $\mathbf{W}_n = 0$, $n \geq T$, can be conceived as the conditional mean with respect to the distribution of underresolved dynamics ($\mathbf{W}_n$, $n \geq T$). We remark that Price et al. [40] also attacked the same problem using an operator algebraic expansion (of time t) to approximate the memory kernel, which is different from our purely data-driven approach. In addition, the GFD (2.11) is guaranteed in our method, but not guaranteed in the t-expansion approximates.

We also tested the standard way of fitting the residual $\mathbf{W}_n$ as independent and multivariate Gaussian distribution (independent between different time indices, and multivariate in the M

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

observables). The results are qualitatively similar but noisier, because the predictions are based on randomly generated samples of the distribution. As the results with the Gaussian noise model are similar but less "clean," we will not present them in this article.

3.3. Linear and nonlinear regression–based projections. We now elaborate a critical idea that distinguishes the subtle differences between two regression-based projections that appear nearly identical. We will refer to the first technique as "linear regression-based projection" and the second as "nonlinear regression-based projection," as we currently lack more precise terminology. Although the regression analyses for these two methods are highly similar, drastic differences exist in the procedures utilized to apply the trained models for making predictions: the resulting reduced-order models are always linear dynamical systems with the first technique but can be nonlinear dynamical systems with the second approach. Consequently, the learned models can even produce qualitatively distinct long-time behaviors, as will be seen in the next section.

To illustrate this idea, we employ a one-dimensional toy model: $\dot{\phi} = -\phi^2 + \phi$ for some initial distribution $\phi(0) \sim \mu(\mathrm{d}\phi)$. We chose $\mu = 0.5 + \mathrm{Beta}(2,2)$ for this example. We generate $N = 10^5$ trajectories in the time window $t \in [0,3]$ and record snapshots every $\Delta = 0.05$ (each trajectory contains $K = 61$ snapshots, including the initial condition). Ten randomly selected trajectories are visualized in Figure 1(a). We will use the generated trajectories to perform two regression analyses, which we will later use for long-term predictions by the truncated GLE (2.5). For simplicity, we will only use the Markov term for making the predictions. This can be achieved by setting $\mathbf{W}_n = 0$ for $n \in \mathbb{Z}_{\geq} 0 = 0$ during the prediction phase. Therefore, we will use the following equation recursively to make a multistep prediction:

$$\mathbf{g}_{m+1}(\boldsymbol{\phi}_0) = \boldsymbol{\Omega}^{(0)}\left(\mathbf{g}_m(\boldsymbol{\phi}_0)\right) \equiv \left[(\mathcal{P}\mathcal{K})^m\, \mathbf{g}\right](\boldsymbol{\phi}_0)\,. \tag{3.11}$$

We would like to emphasize that the concept highlighted in this section is generalizable to systems that include MZ memory. We focus on only the Markov transitions for clarity of presentation.

The first projection is in line with the approximate Koopman learning [44, 52] and data-driven learning of MZ operators with Mori's projection [24]. We will refer to this method as the *linear regression–based projection.* With this method, one first specifies a set of basis functions;

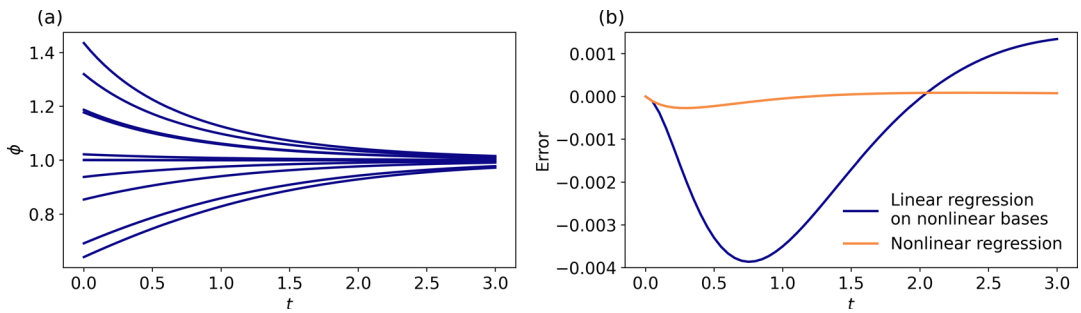

Figure 1. *Learning the toy model* $\dot{\phi} = -\phi^2 + \phi$. (a) *Ten trajectories of the dynamics form randomly* ($\sim 0.5 + Beta(2,2)$) *generated initial conditions.* (b) *The error of the prediction of the two regression-based projection operators.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

we consider polynomial functions up to the quadratic order, $\mathbf{g}(\boldsymbol{\phi}) = [g_1(\phi), g_2(\phi), g_3(\phi)]^T$ with $g_1(\phi) := 1$, $g_2(\phi) =: \phi$, and $g_3(\phi) =: \phi^2$. The data matrix $\mathbf{D}$, whose entries are $D_{i,j,k} = [\phi(k-1; \phi_0^{[i]})]^{j-1}$, is thus $10^5 \times 3 \times 61$. With this method, a regression on the three nonlinear basis functions is used as the projection operator. The linear regression model aims to identify the approximate Koopman operator $\boldsymbol{\kappa} \in \mathbb{R}^{3\times 3}$ that minimizes the MSE:

$$\boldsymbol{\kappa}_*^{(0)} := \underset{\boldsymbol{\kappa}}{\operatorname{argmin}} \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{3} \sum_{k=1}^{60} \left(D_{i,j,k+1} - \sum_{\ell=1}^{3} \boldsymbol{\kappa}_{j,\ell} D_{i,\ell,k} \right)^2, \tag{3.12}$$

where $\boldsymbol{\kappa}_{j,\ell}$ is the (j,ℓ)-component of $\boldsymbol{\kappa}$. The cost function involves all three functional bases, and there are 3×3 parameters in the regression analysis. Note that the three functional bases are treated as *independent variables,* although knowing $g_2(\phi) := \phi$ informs the others for arbitrary ϕ. Our numerical experiment showed that

$$\boldsymbol{\kappa}_*^{(0)} = \begin{bmatrix} +1.000 & +0.000 & -0.000 \\ +0.002 & +1.044 & -0.046 \\ -0.082 & +0.265 & +0.816 \end{bmatrix}. \tag{3.13}$$

With the learned $\boldsymbol{\kappa}_*$, we now make an n-step prediction into the future, provided a current state ϕ_0. The nonlinear function evaluation of the initial state is $\mathbf{g}_0(\phi_0) = \left[\phi_0^0, \phi_0^1, \phi_0^2\right]^T$; here, the subscript of $\mathbf{g}$ represents that the nonlinear functional bases are at time step 0. Because $\mathcal{PK}\mathbf{g} = \boldsymbol{\kappa}_*^{(0)} \cdot \mathbf{g}$ in this regression model, it is straightforward to propagate the functional bases m-steps, $m \geq 1$, into the future to make predictions:

$$\mathbf{g}_m^{(\mathrm{pred})}(\phi_0) = \left(\left(\boldsymbol{\kappa}_*^{(0)} \right)^m \cdot \mathbf{g}_0 \right)(\phi_0), \tag{3.14}$$

that is, directly applying m times the linear transformation $\boldsymbol{\kappa}^*$ to the initial functional bases. Note that this method provides a closed dynamical system of the resolved observables (g_0, g_1, and g_2). Furthermore, the evolutionary equation of the resolved observables is linear. Thus, we can conceive the method as a *linear closure scheme.* Such a procedure for making multistep predictions is natural and commonly adopted in the Koopman framework [2, 27, 52, 53]. The procedure has a drawback: because the prediction of the evolution of the functional bases is linear, the nonlinear moment information is not preserved. For example, the predicted second moment of the system after $m > 0$ steps is not necessarily the squared predicted first moment:

$$\begin{aligned} \left[\mathbf{g}_m^{(\mathrm{pred})}(\phi_0)\right]_3 &= \underbrace{\sum_{\ell=1}^{3} \left(\left(\boldsymbol{\kappa}_*^{(0)} \right)^m \right)_{3,\ell} g_\ell(\phi_0)}_{\text{Predicted second moment}} \\ &\neq \underbrace{\left[\sum_{\ell=1}^{3} \left(\left(\boldsymbol{\kappa}_*^{(0)} \right)^m \right)_{2,\ell} g_\ell(\phi_0) \right]^2}_{\text{Squared predicted first moment}} = \left[\mathbf{g}_m^{(\mathrm{pred})}(\phi_0)\right]_2^2. \end{aligned} \tag{3.15}$$

The above equation (3.14) only uses the learned Markov kernel for making predictions. When we use T MZ memory terms to make a prediction (see section 3.2),

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

$$\mathbf{g}_{n+1}^{(\mathrm{pred})}\left(\boldsymbol{\phi}_{-T}\right)=\sum_{\ell=0}^{T}\boldsymbol{\kappa}_{*}^{(\ell)}\cdot\tilde{\mathbf{g}}_{n-\ell}\left(\boldsymbol{\phi}_{-T}\right),\ n=0,1,\ldots, \tag{3.16a}$$

$$\tilde{\mathbf{g}}_k=\begin{cases}\mathbf{g}_k & \text{if } k=-T,-T+1,\ldots-1,0 \text{ (Given history)},\\ \mathbf{g}_k^{(\mathrm{pred})} & \text{else.}\end{cases} \tag{3.16b}$$

That is, the predicted values are recursively used as imputation when making more than one step of predictions. Because of the linear nature of the projection operator, predictions based on (3.16) are linear in $\{\mathbf{g}_k\}_{k=-T}^{0}$. Below, we shall refer to those predictions based on (3.14) as *linear projection without memory* and (3.16) as *linear projection with memory.*

The second method, which we call the *nonlinear regression–based projection,* is more commonly adopted in modeling. The idea is to use a regression model on a set of independent observables we care about. For those observables which can be determined by (possibly nonlinear) transformation of the independent observables, the explicit form of the transformations is used in the regression model. After learning, one predicts the evolution of the independent observables, using those explicit forms of transformation. With our toy example, there is only one independent observable $\mathbf{g}=[g(\phi)]^T=[\phi]^T$. As such, the data matrix $\mathbf{D}$, with entries $D_{i,1,k}=\phi(k-1;\phi_0^{[i]})$, is $10^5\times 1\times 61$. To match the order of the first method, we consider a second-order polynomial regression as the projection operator. The family of parametric functions is $f_{\boldsymbol{\theta}}(\phi)=f_{\alpha,\beta,\gamma}(\phi):=\alpha+\beta\phi+\gamma\phi^2$ with three fitting parameters $\theta=(\alpha,\beta,\gamma)$. Note that the constant 1 and the ϕ^2 are expressed explicitly as transformation of ϕ. We aim to identify the optimal parameters minimizing the MSE of the one-step prediction

$$\boldsymbol{\theta}^*:=\underset{\boldsymbol{\theta}}{\operatorname{argmin}}\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{60}\left[D_{i,1,k+1}-f_{\boldsymbol{\theta}}\left(D_{i,1,k};\theta\right)\right]^2. \tag{3.17}$$

The above optimization problem, (3.17), is a subproblem of (3.12). As such, the solution of (3.17) is identically the second row of the solution (equation (3.13)): $\alpha^*=\kappa_{1,1}^*\approx 0.002$, $\beta^*=\kappa_{1,2}^*\approx 1.044$, and $\gamma^*=\kappa_{1,3}^*\approx -0.046$. Despite an identical optimization solution, the prediction based on this nonlinear regression model is distinct from the first method (equation (3.14)) because $\boldsymbol{\Omega}^{(0)}(g)=\mathcal{P}\mathcal{K}g=\alpha_*^{(0)}+\beta_*^{(0)}g+\gamma_*^{(0)}g^2$ now:

$$\mathbf{g}_m^{(\mathrm{pred})}(\phi_0)=\underbrace{f_{\boldsymbol{\theta}_*^{(0)}}\left(\ldots\left(f_{\boldsymbol{\theta}_*^{(0)}}\left(\mathbf{g}_0\left(\boldsymbol{\phi}_0\right)\right)\right)\right)}_{m \text{ function compositions}}. \tag{3.18}$$

Such a recursive composition of a nonlinear mapping is more in line with most modeling for nonlinear dynamical systems: we assume a structure of the vector field, and after fitting the free parameters, the best-fit model is treated as a nonlinear dynamical system (versus a linear system of an augmented set of nonlinear functional bases, (3.14)). Nonlinear systems identification methods, such as sparse identification of nonlinear dynamics (SINDy [5, 12, 17]), also follow the recursive equation (3.18) in the prediction phase. Note that this method also provides a closed dynamical system of the resolved observables, but in this case there is only one resolved observable, $g(\phi)=\phi$, yielding a nonlinear evolutionary equation of ϕ. Below, we refer to this method as a *nonlinear closure scheme.*

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Analogous to (3.14), (3.18) only uses the learned Markov model for making predictions. To make a prediction with T MZ memory terms,

$$\mathbf{g}_{n+1}^{(\text{pred})}\left(\boldsymbol{\phi}_{-T}\right) = \sum_{\ell=0}^{T} \mathbf{f}_{\theta_*^{(\ell)}}\left(\tilde{\mathbf{g}}_{n-\ell}\left(\boldsymbol{\phi}_{-T}\right)\right), \; n = 0, 1, \ldots, \tag{3.19a}$$

$$\tilde{\mathbf{g}}_k = \begin{cases} \mathbf{g}_k & \text{if } k = -T, -T+1, \ldots -1, 0 \text{ (Given history)}, \\ \mathbf{g}_k^{(\text{pred})} & \text{else.} \end{cases} \tag{3.19b}$$

Similar to (3.16), predicted values are recursively used as imputation when making more than one step of predictions. With the linear nature of the projection operator, predictions based on (3.16) are no longer linear in $\{\mathbf{g}_k\}_{k=-T}^{0}$. Below, we shall refer to those predictions based on (3.18) as *nonlinear projection without memory* and (3.19) as *nonlinear projection with memory.*

To illustrate the difference between these two closure schemes, we used the learned $\boldsymbol{\kappa}_*^{(0)}$ and $(\alpha_*^{(0)}, \beta_*^{(0)}, \gamma_*^{(0)})$ to predict the dynamics of the toy model every $\Delta = 0.05$ up to $t = 3.0$, using linear projection without memory (equation (3.14)) and nonlinear projection without memory (equation (3.18)). In this demonstration, we set the initial condition of the test trajectory at $\phi(0) = 1.4$. The error of the prediction to the true dynamics is illustrated in Figure 1(b), which shows that the nonlinear closure scheme performed better than the linear one. In the next section, we will see that predictions based on the linear closure scheme could even lose important qualitative features of the fully resolved dynamics.

The nonlinear projection is a better formulation for accommodating data-driven models with neural networks, even in the context of approximate Koopman learning [44, 52] and data-driving learning of MZ operators with Mori's projection [24]. For these problems, the set of observables have to be specified a priori. Because the identification of the optimal basis functions remains an open problem, several recently proposed methods leverage neural networks for learning better observables [22, 27, 51]. However, neural networks are nonlinear functions of their parameters, so the joint learning problem (i.e., simultaneously learning the functional basis and the approximate Koopman operator), often with additional nonlinear regularizers, is not a linear regression problem. Consequently, we cannot categorize the neural network–based approach as a simple Mori's projection, for which our proposed algorithms [24] can extract the MZ memory kernels. A projection defined by the nonlinear projection is the most natural and accurate mathematical framework to describe these neural network–based learning methods. By identifying "training a neural network" as a projection, the algorithm described in section 3.1 prescribes a principled way to extract the memory kernels of these neural network–based learning methods.

3.4. A special case: Linear regression. We now show that the special choice of linear projection results in the discrete-time algorithm we proposed in [24]. For any observation $\mathbf{g}(\boldsymbol{\phi}_0)$, linear regression postulates that the parametric form of $\mathbf{f}$ is a linear superposition of the independent variables:

$$\mathbf{f}\left(\mathbf{g}\left(\boldsymbol{\phi}_0\right); \boldsymbol{\kappa}\right) := \boldsymbol{\kappa} \cdot \mathbf{g}\left(\boldsymbol{\phi}_0\right). \tag{3.20}$$

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Generally, we include the constant function that maps any $\boldsymbol{\phi}$ to 1 in the vectorized observable $\mathbf{g}$ to account for the biased term in linear regression analysis. Here, the parameter $\boldsymbol{\theta}$ is the list of entries of the $M \times M$ matrix $\boldsymbol{\kappa}$. The best fit of a linear regression is analytically tractable. Define the $N \times M$ data matrix $\mathbf{X}$ at time $k-1$ whose entries are $\mathbf{X}_{i,j}(k-1) := \mathbf{D}(i,j,k)$. The solution of (3.2) is

$$\boldsymbol{\kappa}_*^{(0)} = \mathbf{C}(1) \cdot \mathbf{C}^{-1}(0), \tag{3.21}$$

where $\mathbf{C}(k) := \mathbf{X}^T(k) \cdot \mathbf{X}(0)$ is the empirical k-lag correlation matrix of the observations. Equation (3.21) is precisely the extracted Markov term in [24], and the approximate Koopman operator [44, 45, 52]. As for the higher orders, (3.8a) can be expressed as

$$\mathbf{y}_{n+1} := \mathbf{X}(n+2) - \sum_{\ell=0}^{n} \boldsymbol{\kappa}_*^{(\ell)} \cdot \mathbf{X}(n-\ell+1), \tag{3.22}$$

leading to the solution of the minimization problem (3.8b)–(3.8d),

$$\boldsymbol{\beta}_*^{(n+1)} = \left[\mathbf{C}(n+2) - \sum_{\ell=0}^{n} \boldsymbol{\kappa}_*^{(\ell)} \cdot \mathbf{C}(n-\ell+1) \right] \cdot \mathbf{C}^{-1}(0), \tag{3.23}$$

which is exactly the formula derived from the GLE with Mori's projection operator [24]. Thus, the linear regression–based projection with memory is equivalent to Mori's and the finite-rank projection. In addition, the Markov transition kernel with the linear regression–based projection is the approximate Koopman operator learned by EDMD [24, 52]. As such, linear projection without memory is equivalent to the approximate Koopman analysis such as DMD [44] and EDMD [52], and the linear projection with memory is equivalent to our proposed data-driven learning with Mori's projection [24].

4. Numerical experiments.

4.1. An illustrative example: The Van der Pol oscillator. We now use the Van der Pol oscillator as an example to illustrate the regression-based learning of MZ operators in action. The Van der Pol oscillator is a two-dimensional nonlinear system, which follows

$$\frac{\mathrm{d}}{\mathrm{d}t}\varphi(t) = \mu\left(\varphi - \frac{\varphi^3}{3}\right) - \psi, \tag{4.1a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\psi(t) = \frac{1}{\mu}\varphi. \tag{4.1b}$$

We chose $\mu = 1$ in this illustrative example. The Van der Pol oscillator has a globally stable limit cycle, which is a one-dimensional manifold embedded in the $\boldsymbol{\phi} = (\varphi, \psi)$-state space, as shown in Figure 2(a). For illustrative purposes, we considered a single observable $g(\boldsymbol{\phi}) \triangleq \varphi$ as the resolved dynamic variable. We used the `lsoda` integrator in `scipy.integrate.solve_ivp` to numerically simulate the two-dimensional system from an arbitrarily chosen initial state $\boldsymbol{\phi}_0 = (0,1)$ and a relaxation time $t_c = 100$ for the system to relax to the limit cycle, whose

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

period is roughly 6.663. We chose 50 initial conditions evenly sampled over one period; these initial conditions are also visualized in Figure 2(a). Then, we collected snapshot data set $\mathbf{D}$ by measuring $\varphi(t)$ from $t=0$ to $t=20$ every $\Delta=0.5$ from each of the initial conditions; two of the trajectories are shown in Figure 2(b). We used the MSE as the cost function for the regression models below. In Figures 2(c)–(g), we show how Mori's projection operator (i.e., linear projection) is applied to the snapshot data set $\mathbf{D}$. In Figure 2(c), the paired samples are simply $x=\mathbf{D}(\cdot,1,1)$ and $y=\mathbf{D}(\cdot,1,2)$. That is, the x's are the observed φ at the time $t=0$, and the $y^{(0)}$'s are the observed φ one step ahead at the time $t=\Delta$. Because the system is not fully resolved, there can be two possible y's for most of the measured x on the stable limit cycle (see Figure 2(a)). Approximate Koopman and Mori's projector on a single observable is equivalent to a linear regression using $f(x;\theta)=\theta x$ to fit the data. The best-fit line is the projected model, $\theta_*^{(0)}x$. The slope of the best-fit model, $\theta_*^{(0)}$, is the approximate Koopman operator [44, 45, 52] and equivalently the 1×1 Markov transition matrix in the context of data-driven MZ formalism [24]. After the first regression is made, the learned model $\theta_*^{(0)}$ is used to predict the system from $t=\Delta$ to 2Δ, and the residual $\mathbf{D}(\cdot,1,3)-\theta_*^{(0)}\mathbf{D}(\cdot,1,2)$ is assigned to $y^{(1)}$. The next linear regression on the independent variable, which remains as $x=\mathbf{D}(\cdot,1,1)$, is performed (Figure 2(d)), and the slope of the best-fit line is the first discrete-time 1×1 memory kernel [24]. The process repeats, and in Figures 2(e)–(g), we showed the 5th, 10th, and 30th linear regression. In Figures 2(h)–(l), we used fifth-order nonlinear polynomial regression instead. As expected, the more expressive polynomial regression led to smaller errors than that of the linear regression.

Figures 2(m)–(o) show the predictions of the learned models. First, we simulated the full system (4.1) from a different initial condition $\boldsymbol{\phi}_0=(1,0)$ until $t=t_c=100$ for relaxing the system to the stable limit cycle. Then, we measured $\varphi(t+t_c;\boldsymbol{\phi}_0)$ every $\Delta=0.5$ until $t=260$ as the ground-truth of the dynamics. We considered three learned models that are more expressive than a single linear observable (ϕ), referred to as (1) linear projection with memory (order 1): two polynomial functions as the observables, $\mathbf{g}(\boldsymbol{\phi})\triangleq[1,\varphi]^T$, predicted by (3.16); (2) linear projection with memory (order 5): six lowest-order polynomial functions as the observables, $\mathbf{g}(\boldsymbol{\phi})\triangleq\left[1,\varphi,\varphi^2,\dots\varphi^5\right]^T$), predicted by (3.16); (3) nonlinear projection with memory (order 5): six lowest-order polynomial functions as the observables, $\mathbf{g}(\boldsymbol{\phi})\triangleq\left[1,\varphi,\varphi^2,\dots\varphi^5\right]^T$), predicted by (3.19). Given the snapshots in $t\in(100,120)$, we used (3.16) or (3.19) to predict the dynamics at $t\geq 120$. Figures 2(m) and (n) show that the error due to the negligence of the orthogonal dynamics accumulates and eventually destabilizes the prediction in Mori's projection formalism. In contrast, Figure 2(o) shows that the nonlinear regression–based projection operator can reasonably approximate the true dynamics for a longer predictive horizon, using only the resolved snapshots.

We remark that this example was deliberately constructed as a minimal example. Our intention is to use a simple system and its transparent visualizations (Figure 2) to illustrate the iterative learning of the MZ operators in action.

4.1.1. Lorenz (1963) system. Our second example is the Lorenz (1963) model [25], which is a three-dimensional system,

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

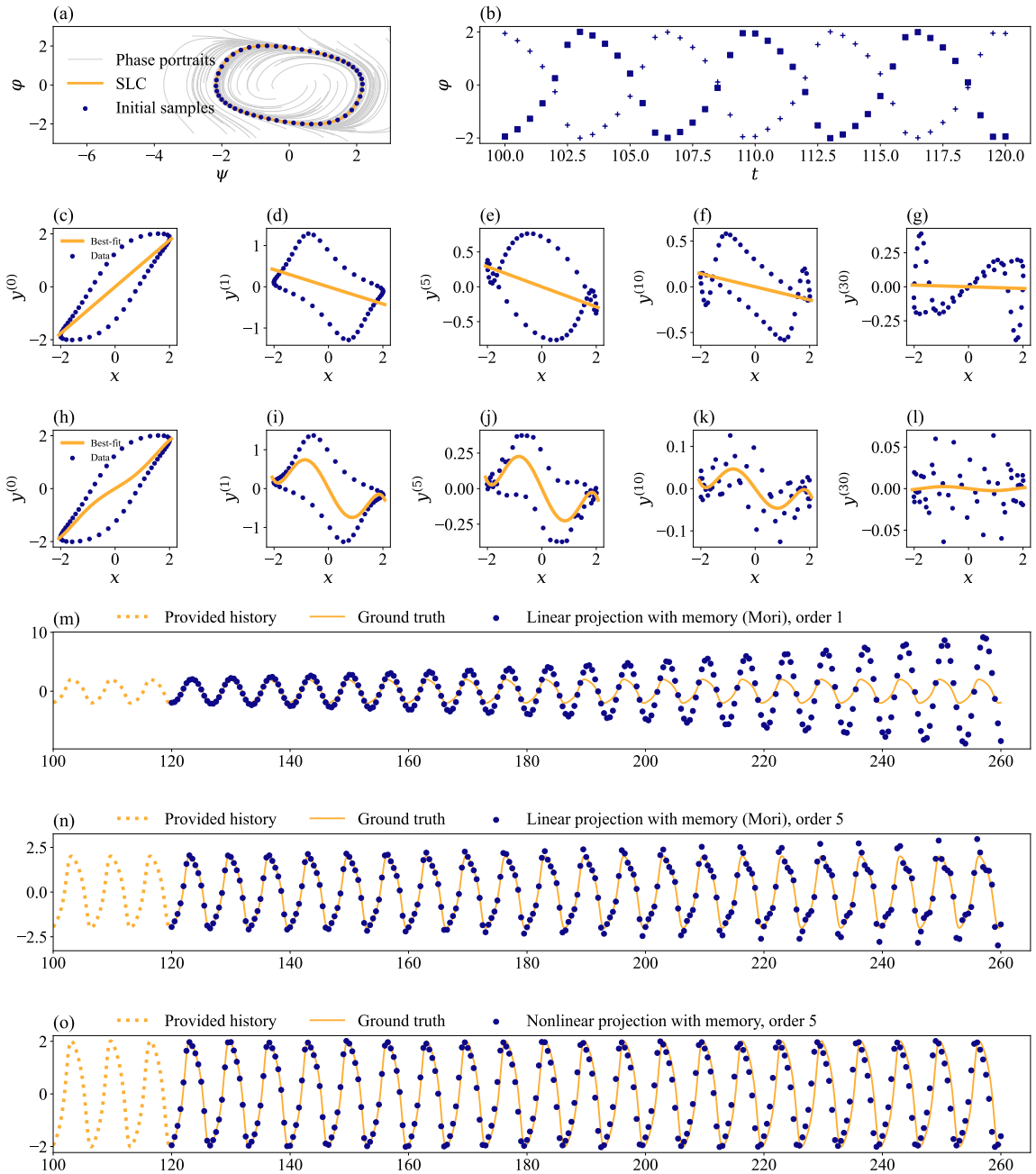

Figure 2. *The Van der Pol oscillator.* (a) *The phase portraits of the full system, the stable limit cycle, and the initial samples of the data matrix* $\mathbf{D}(\cdot, 1, 1)$. (b) *Snapshot samples along two trajectories,* $\mathbf{D}(1, 1, \cdot)$ *and* $\mathbf{D}(26, 1, \cdot)$. (c)–(g) *Linear regression with a single observable,* $f(x; \theta) := \theta x$. (h)–(l) *Nonlinear polynomial regression with a single observable,* $f(x; \boldsymbol{\theta}) := \sum_{i=0}^{5} \theta_i x^i$. (m)–(o) *Prediction by* (m) *linear projection with memory (equation* (3.16)*) on* $\mathbf{g}(\boldsymbol{\phi}) \triangleq [1, \varphi]^T$*;* (n) *linear projection with memory (equation* (3.16)*) on* $\mathbf{g}(\boldsymbol{\phi}) \triangleq [1, \varphi, \dots \varphi^5]^T$*; and* (o) *nonlinear projection with memory (equation* (3.16)*) on* $\mathbf{g}(\boldsymbol{\phi}) \triangleq [1, \varphi, \dots \varphi^5]^T$.

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

$$\frac{\mathrm{d}}{\mathrm{d}t}\varphi(t) = \sigma(\psi - \varphi), \tag{4.2a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\psi(t) = \varphi(\rho - \chi) - \psi, \tag{4.2b}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\chi(t) = \varphi\psi - \beta\chi. \tag{4.2c}$$

We adopted the original model parameter values that Lorenz chose, $(\sigma, \rho, \beta) = (10, 28, 8/3)$. Note that the full phase-space state $\boldsymbol{\phi}$ is $[\varphi, \psi, \chi]$. We again considered the "reduced-order model" as the $\varphi(t)$ alone. A long trajectory with an arbitrarily selected initial condition $\boldsymbol{\phi}_0^{\mathrm{tr}} = (0.01, 1, 10)$ was generated using `scipy.integrate.solve_ivp` for data-driven learning. We discarded the initial transient before $t = t_c = 1000$, and collected 10^6 snapshots of φ every $\Delta = 0.01$ until $t = t_c + 10^4$, as the samples on the strange attractor. A test trajectory was generated from a different initial condition $\boldsymbol{\phi}_0^{\mathrm{te}} = (0, 1, 2)$, and also with $t_c = 1000$. Again, we used the MSE as the cost function for the regression models.

We considered five projection operators and compared their performance, ordered by the complexity (expressiveness) of the statistical model below. First, we again considered the order-1 linear projection operator, i.e., linear regression on the vectorized observable defined by $\mathbf{g}(\boldsymbol{\phi}) = [1, \varphi]^T$. We will refer to this as the Mori (order-1). Without the memory effects, such a projection operator corresponds to the DMD analysis [44, 45]. Next, we considered the order-5 linear projection operator but on five polynomial basis functions, that is, linear regression on the vectorized observable which maps the full state $\boldsymbol{\phi}$ to $\left[1, \varphi, \varphi^2 \dots \varphi^5\right]^T$. We will refer to this as the Mori (order-5). Without the memory, such a projection operator corresponds to an EDMD analysis [52]. The third projection operator we considered is an order-5 nonlinear projection defined as the fifth-order polynomial regression on φ. We will refer to this as the polynomial regression (order-5). Next, we considered a nonlinear projection operator defined by a cubic (degree $= 3$) spline regression with 10 knots (including boundary knots), evenly distributed between 1.5 times of the minimum and maximum of the collected φ samples. After the spline features are generated, we performed a ridge regression with the L^2-regularization parameter set at $\lambda = 10$. Finally, we considered a simple neural network as the nonlinear projection operator. The architecture of the neural networks is two fully connected feed-forward layers, each of which contains five artificial neurons, and a third linear layer with only one neuron. In this study, we always chose the hyperbolic tangent function as the activation function of the artificial neurons.

We used the measured snapshots to learn the operators $\boldsymbol{\Omega}^{(\ell)}$ for each of the selected projection operators. Figure 3(a) shows the magnitude of the summary residual, defined as the averaged MSE of the best ℓth regression model on the test trajectory. Recall that ℓ is the index of the memory terms (cf. (2.5)). Thus, we plotted the summary residual as a function of the physical time of the memory length ($\ell\Delta$) in Figure 3(a). In addition, for linear projections, the full optimization problem is the sum of the MSE of all the observables, in contrast to only the MSE of φ in those nonlinear projections. To make a consistent comparison, we only quantified and reported the MSE of the component φ for linear projections in the figure. It is observed that with the increased complexity of the regression model, the residual with memory contribution can decrease significantly, from order-1 linear projection ($\approx 2 \times 10^{-1}$) to the expressive neural network ($\approx 2 \times 10^{-2}$). Note that without the memory contribution ($\ell = 0$),

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

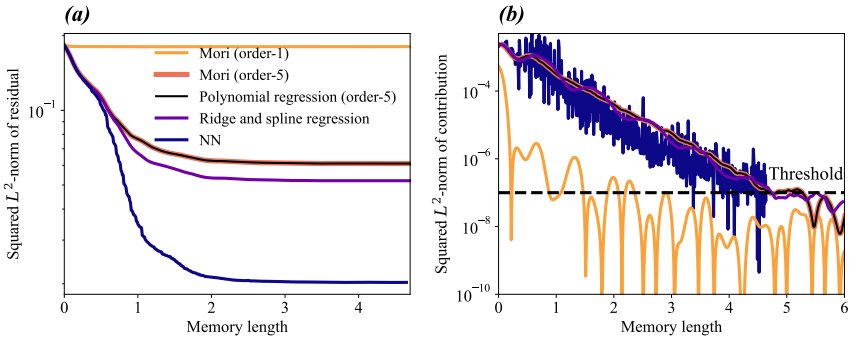

Figure 3. *The Lorenz* (1963) *system.* (a) *The squared* L^2*-norm of the validation residual as the memory length increases.* (b) *The squared* L^2*-norm contribution at different memory length. The threshold is set at* 10^{-7} *for determining the memory length. Note that in these visualizations, Mori* (5) *and nonlinear* (5) *are identical. The memory length is in the physical time unit, i.e., memory length* 1 *corresponds to* 100 *discrete-time steps* ($\Delta = 0.01$).

the magnitudes of the summary residual are indistinguishable; the significant improvement comes only with a finite-time memory contribution, ≈ 2.0. This observation justified the advantage of Mori and Zwanzig's memory-dependent formulation, especially for more expressive models.

To investigate the contribution of the memory effects, we plotted the L^2-norm of the memory contribution in (2.5) $(\Omega^{(\ell)}\left(\boldsymbol{\phi}\left(k\Delta + t_c; \boldsymbol{\phi}_0^{\text{te}}\right)\right),\ k = 0, 1, \ldots K + N)$ averaged over the collected snapshots of the test trajectory. That is, we chose each snapshot along the long trajectory as a sample of the initial condition, and computed the memory contribution ℓ steps into the future. For order-1 linear projection operators, we only quantified the memory contributions to φ, i.e., $[\boldsymbol{\Omega}^{(\ell)} \cdot \boldsymbol{\phi}\left(k\Delta + t_c; \boldsymbol{\phi}_0^{\text{te}}\right)]_2$ (note that the first component is the constant function 1). Figure 3(b) shows the memory contribution with various projection operators. We observed that memory contributions decayed as the memory index ℓ increased, regardless of the regression method. This observation suggests that a finite length (H) of the memory is sufficient, as the contribution of past history after a certain timescale would be small and negligible. As a result, in the following analysis, we chose a finite memory contribution by thresholding the squared L^2-norm of the memory contribution at 10^{-7}. The corresponding memory lengths are $H = 235$ snapshots for order-1 linear projection; $H = 469$ snapshots for order-5 linear projection; $H = 469$ snapshots for order-5 polynomial regression; $H = 469$ snapshots for spline regression; $H = 469$ snapshots for the neural network. We note that the number of discrete-time memories is much higher than what would have been needed for time-embedding techniques, such as Takens delay embedding [48]. This provides the numerical evidence that the MZ memory is not the same memory contribution in the Takens delay-embedding technique, a point that we will elaborate in section 5.

After the memory kernels ($\boldsymbol{\Omega}^{(\ell)}$) are learned, we turned our attention to the prediction of the learned models. We took the past history of the reduced-order variable $\varphi(1000 \le t \le 1050)$ and use (3.9) to make predictions for $t > 1050$, assuming the orthogonal dynamics $\mathbf{W}_n = 0$. For comparison, we also made a prediction from memoryless models, that is, setting $\boldsymbol{\Omega}^{(\ell \ge 1)} = 0$. Note that the order-1 and order-5 linear without memory corresponds to DMD [44] and EDMD [52], respectively.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Figure 4 shows the predicted trajectories by the learned models, using (3.16) for linear projections and (3.19) for nonlinear projections. For all the methods, we found that the memoryless models failed to capture the characteristics of the resolved dynamics other than the long-time mean. For order-1 and order-5 linear projections, a similar behavior of relaxation to the mean was observed. This should not be a surprise: as detailed in our recent work [24], linear projections without the unresolved orthogonal dynamics is functionally identical to using the two-time correlation functions as the propagators for prediction. The two-time correlation

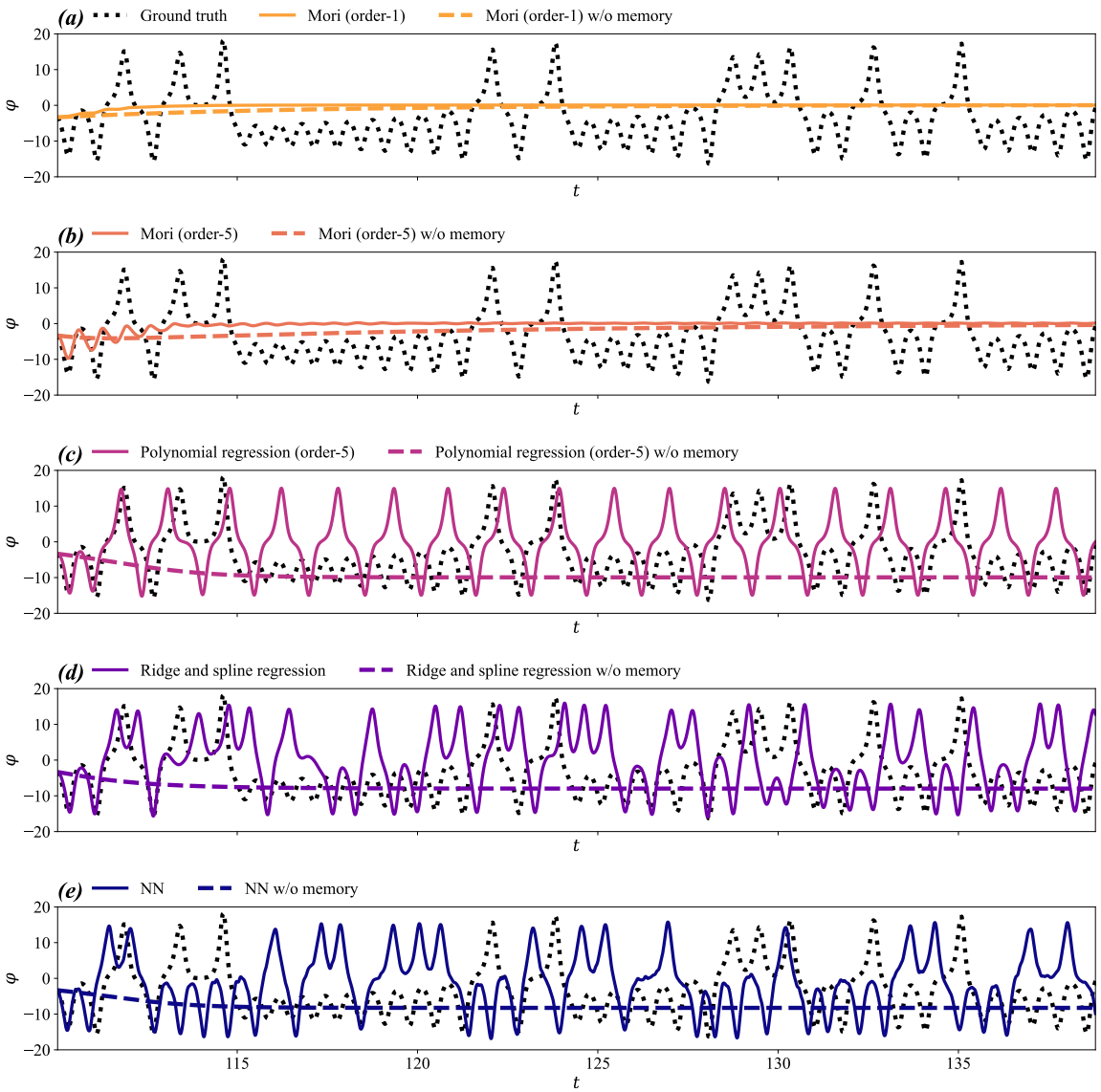

Figure 4. *The Lorenz* (1963) *system. The predicted trajectories for the learned models based on* (a) *Mori* (1) *projection operator,* (b) *Mori* (5) *projection operator,* (c) *a projection by fifth-order polynomial regression on* φ*,* (d) *a projection by ridge and spline regression, and* (e) *a projection by a neural network .*

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

function $\mathbf{C}_{i,j}(t)$ decays to 0 because the chaotic dynamics decorrelates at a large lag $t \gg 1$. Consequently, linear projections also made predictions that converge to the mean behavior in the long-time limit. They were able to predict the mean because a constant function 1, which can capture the constant bias, was included in the set of observables. Notably, linear projections with the memory kernels learned to predict a transient oscillation which crudely characterized the short-time dynamics of the Lorenz (1963) model. The drawback of the steady long-time prediction was significantly improved when we used nonlinear projection. The fifth-order polynomial regression showed a much better prediction in a short horizon, and an oscillation between the two wings of the Lorenz butterfly. Notably, the oscillation is not chaotic as it is in the fully resolved system. The prediction based on the spline regression, which is an intermediate model between the nonlinear polynomial regression and the neural network–based regression models, began to exhibit chaotic behaviors in the long-time limit. Finally, the prediction of the one-dimensional reduced-order model based on a neural network was qualitatively similar to the chaotic dynamics of the fully resolved Lorenz (1963) model.

To quantify the performance of the predictions, we first collected the long-time statistics by evolving (3.9) for 150,000 steps. We plotted the distributions of the collected trajectories based on the learned models in Figure 5; such empirical distributions would be the stationary distribution if the process is ergodic. Clearly, linear projections showed an almost δ-like distribution at $\varphi = 0$ because their predictions converged to the mean in a finite timescale. All the predictions based on nonlinear projections reasonably approximated the empirical distribution of φ. We observed that the neural network outperformed the spline regression, which outperformed the plain fifth-order polynomial regression.

We further quantified the prediction error by the following metrics. First, we computed the squared L^2-error of the prediction to the actual dynamics of $\varphi(t)$. The initial condition and the histories were uniformly sampled along a long trajectory of the actual dynamics. We collected 25,000 samples and computed the MSE as a function of the predictive horizon in Figure 6(a). Within a short horizon, predictions by the nonlinear projections were better than linear projections. The errors of the nonlinear projections eventually grew larger at a finite predictive horizon ($t \approx 0.3$). We remark that this should be considered not as a drawback but as an advantage: beyond this timescale, linear projection predicted the mean of the chaotic dynamics, but nonlinear projection predicted oscillatory solutions. In terms of the MSE, predicting only the mean would have a smaller error. As such, we also computed the Kullback–Leibler divergence from the predictive distribution (approximated by the histogram of the samples from of the reduced-order models) to the ground-truth distribution (approximated

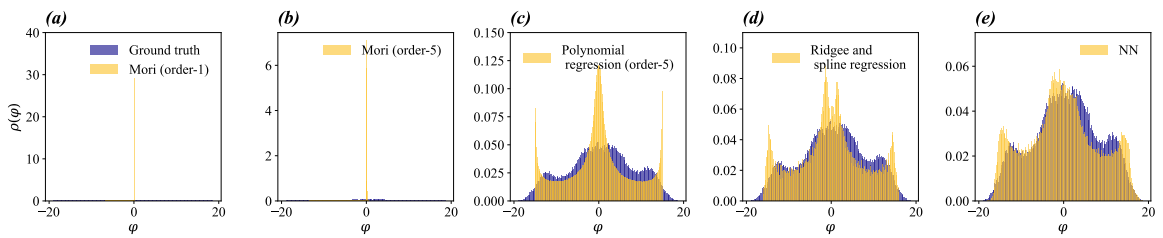

Figure 5. *Long-time statistics of the MZ model predictions of the Lorenz* (1963) *model.*

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

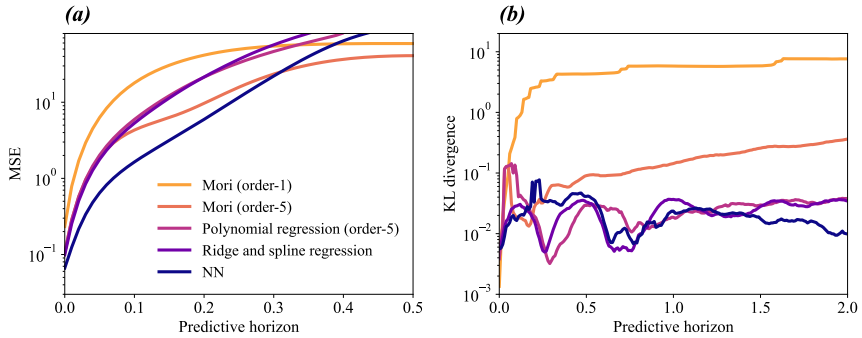

Figure 6. *Error of the reduced-order models on the Lorenz* (1963) *system.* (a) *The MSE of the prediction as a function of the predictive horizon;* (b) *the Kullback–Leibler divergence of the empirical distribution from the model predicted distribution as a function of the predictive horizon.*

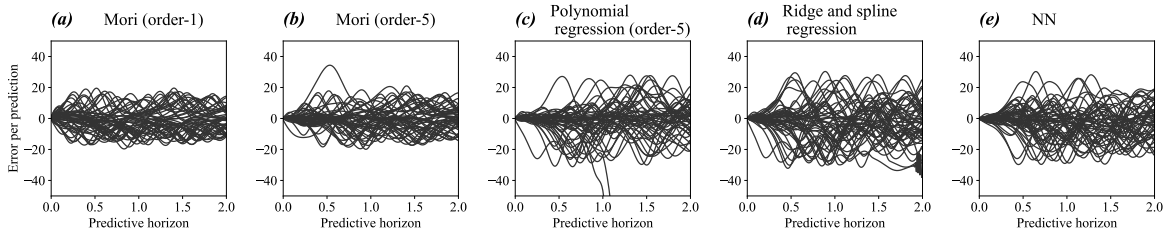

Figure 7. *The Lorenz* (1963) *system.* (a)–(e) *The deviation of* 50 *predicted trajectories using the learned models.*

by the histogram of the samples from the actual dynamics), as shown in Figure 6(b). With this metric, we observed that the reduced-order models based on the nonlinear projection operators captured the time-dependent distribution of φ much better than the reduced-order models based on linear projection operators.

We conclude this section by discussing the potential pitfalls of a regression-based nonlinear projection operator. Mainly, the stability of the learned models is not always guaranteed. We observed that the learned reduced model, provided some initial histories, can blow up in finite time. In Figure 7, we plot the deviation of the prediction from the actual dynamics. For each of the reduced-order models, we plotted 50 deviations, whose initial condition and histories were sampled on a long ground-truth trajectory. We observed that the prediction for the plain fifth-order polynomial regression could explode in finite time. The spline regression, which predicts constants outside the boundary knots, seemed to alleviate the problem. However, when the prediction goes beyond the data distribution, the spline regression can also be "trapped" locally and induce high-frequency oscillations. Interestingly, we did not observe these pathologies in the neural network.

4.2. Kuramoto–Sivashinsky equation. Our third example is the Kuramto–Sivashinsky equation, which was developed in the 1970s for modeling instability of flame propagation [21, 46, 47]. Specifically, in this study, we considered the one-dimensional Kuramto–Sivashinsky (partial differential) equation

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

$$0 = \frac{\partial}{\partial t}u(x,t) + \frac{\partial^2}{\partial x^2}u(x,t) + \frac{\partial^4}{\partial x^4}u(x,t) + u(x,t)\frac{\partial}{\partial_x}u(x,t), \tag{4.3a}$$

for all time $t \geq 0$ on a bounded domain $x \in [0, L]$, given initial data $u(x,0) = u_0(x)$. A periodic boundary condition was imposed, $u(t,0) = u(t,L)\ \forall t$. We chose $L = 16\pi$ and uniformly discretized the space into $N = 128$ points. We used the pseudospectral fourth-order exponential time-derivative Runge–Kutta (ETDRK4) algorithm [18] with 2/3-dealiasing [23] to solve the spatially discretized system. The time step for the explicit ETDRK4 algorithm was set as 10^{-3}, and we collect the snapshots every 10^3 steps (so $\Delta = 1$). For the training data, we followed [18] and set $u_0^{\mathrm{tr}}(x) = \cos(2\pi x/L)[1 + \sin(2\pi x/L)]$, and for the test data, $u_0^{\mathrm{te}}(x) = \sin(2\pi x/L)[1 + \cos(2\pi x/L)]$. Both the training and the validation sets were simulated to $t = t_c + 10^5$, and we discarded the transient trajectories before $t_c = 5 \times 10^2$. For reference, the Lyapunov time of the system is ≈ 12.56 [11]. We still chose MSE as the cost function for the regression models.

We considered a fourfold reduced-order model by only observing the variable u every four discretized spatial grids. That is, $u(t, x_{4i})_i$, where $i = 0, 1, \ldots 31$ is the grid index. We augmented the reduced-order data set by two operations: (1) *shifting*: we used samples collected on different subgrids, $u(t, x_{4i+j})$, $j = 0, 1, 2, 3$; and (2) *reordering*: we imposed the C_1 symmetry of the system, i.e., $u\left(t, x_{\mathrm{mod}(4(i+k)+j,32)}\right)$, $k = 0, 1, \ldots 31$.

We also adopted and integrated the delay-embedding technique into some of the regression-based models. With the delay embedding, the input of the regression analysis is augmented to include a finite number, $E \in \mathbb{Z}_{\geq 0}$, of the current and past snapshots [3, 27]. As noted in the discussion in our previous study [24], the history dependence in the delay-embedding technique is not the memory effect in the MZ formalism. We will provide a more detailed discussion later. For the rest of this article, we refer to the former as "delay embedding" and the latter as "(MZ) memory effect," "(MZ) memory correction," or simply "memory" in the figures. Our major aim for the analysis is to compare the performance of the models with delay embedding and without MZ memory correction to those models with MZ memory correction but without delay embedding.

We considered the following regression-based models and compared their performance. The first model is the linear Mori projection operator. Our preliminary analysis showed that a direct application of either data-driven Koopman [44] or Mori's [24] on the snapshot observations $\{u(\cdot, x_{4i})\}_{i=0}^{31}$ performed poorly (data not reported). Thus, we adopted the delay-embedding technique [3, 27] to augment the basis functions for learning the MZ kernels based on Mori's projection operator (i.e., linear projection with memory). The model will be referred to as the "Mori+DEm" model. Specifically, we use the past E snapshots to augment the input vector (which lives in $\mathbb{R}^{32E}$). For Mori's projection operator, we chose a long embedding $E = 10$. We remark that the lowest-order result of the Mori's projection operator (i.e., without the higher-order memory corrections, $\mathbf{\Omega}^{(\ell)}$, $\ell \geq 1$, in (2.5)) is functionally identical to the Hankel-DMD [3], which is a method solely based on delay embedding. The second regression-based model was chosen to be an FCNN. The architecture of the neural network was two fully connected layers, each of which contained 32 artificial neurons, and a third linear layer with 32 outputs. The third regression-based projection operator was a CNN. CNNs are translationally invariant; such inductive bias is considered as beneficial for systems like the

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Kuramto–Sivashinsky equation, which are invariant under translations. The architecture of the CNN was two one-dimensional convolutional layers with five channels and kernel size of 11. The circular padding was implemented to impose the periodic boundary condition. As the fourth projection operator, we adopted the delay-embedding technique and used $E = 4$ past snapshots as the input for a CNN model. In practice, the past E snapshots were stacked in the channel dimension of the input tensor of CNN. The architecture of the delay-embedded CNN model is identical to the one without delay embedding. This model will be referred to as the "CNN + DEm" model.

We used the snapshots collected along the long trajectory with the initial condition u_0^{tr} to learn the MZ kernels $\mathbf{\Omega}^{(\ell)}$ for each regression-based projection operator. For linear projection operators and for FCNN, we applied both shifting and reordering data augmentation to impose the symmetries. For CNNs, we applied only the shifting data augmentation. We sampled from the generated test trajectory to perform error analyses of the learned regression-based MZ kernels.

As a baseline reference for comparison, we performed reduced-order simulations. In these simulations, the reduced-order snapshots ($\{u_{t,x_{4i}}\}_{i=0}^{31}$) were treated as the full state of the Kuramto–Sivashinsky equation discretized on 32 spatially discretized grids (versus 128 of the ground-truth data-generation process). Each of the snapshots was evolved forward in time directly by the ETDRK4 integrator.

In Figure 8, we visualize the predictive trajectories and errors from the learned models, each with $\ell \leq 10$ MZ memory kernels. We chose $H = 10$ and used the truncated GLE (3.9) for making the predictions, again with the assumption $\mathbf{W}_n = 0$. Figures 8(b) and (c) show the spatiotemporal evolution of the reduced-order simulation. Qualitatively speaking, we observe that the reduced trajectory deviated from the "ground-truth" at a prediction horizon ≈ 20, but it is capable of reproducing dynamical features that are visually indistinguishable. Mori's projected model with $E = 10$ delay embedding (Figure 8(d), (e)) quickly converged to the mean of the dynamics, as we expected. The FCNN model deviated from the ground-truth trajectory sooner than the reduced simulation did, but it also reproduced similar dynamical features; see Figure 8(f), (g). Predictions from both CNN models deviated later than the reduced simulation did (Figure 8(h)–(k)). The $E = 4$ delayed-embedded CNN (Figure 8(j), (k)) did not significantly improve the plain CNN (Figure 8(h), (i)).

Quantitative error analysis of the regression models were performed with 15,000 trajectories uniformly sampled on the generated test trajectory. Figure 9(a) shows the normalized residual of the one-step prediction, defined as the MSE of the ℓth-order MZ model, normalized by that of the Markovian model (i.e., only $\mathbf{\Omega}^{(0)}$). The absolute error values are presented in the inset of Figure 9(a). For all the regression models, including the MZ memory reduced the prediction error. We observed that the MZ memory effects can be captured with less than 10 memory kernels, justifying our choice of $H = 10$ for the predictive model. The plain CNN model without delay embedding showed the highest percentage improvement by including the MZ memory. The delay-embedded CNN model had a slightly smaller percentage improvement in comparison to that of the plain CNN model, although it has smaller absolute magnitude of MSE. The smaller absolute error of the $E = 4$ delay-embedded model is expected, as it is a more expressive model due to its larger input space, allowing coupling the delayed observables. Surprisingly, the plain CNN with $H = 4$ MZ contribution has a smaller absolute error

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

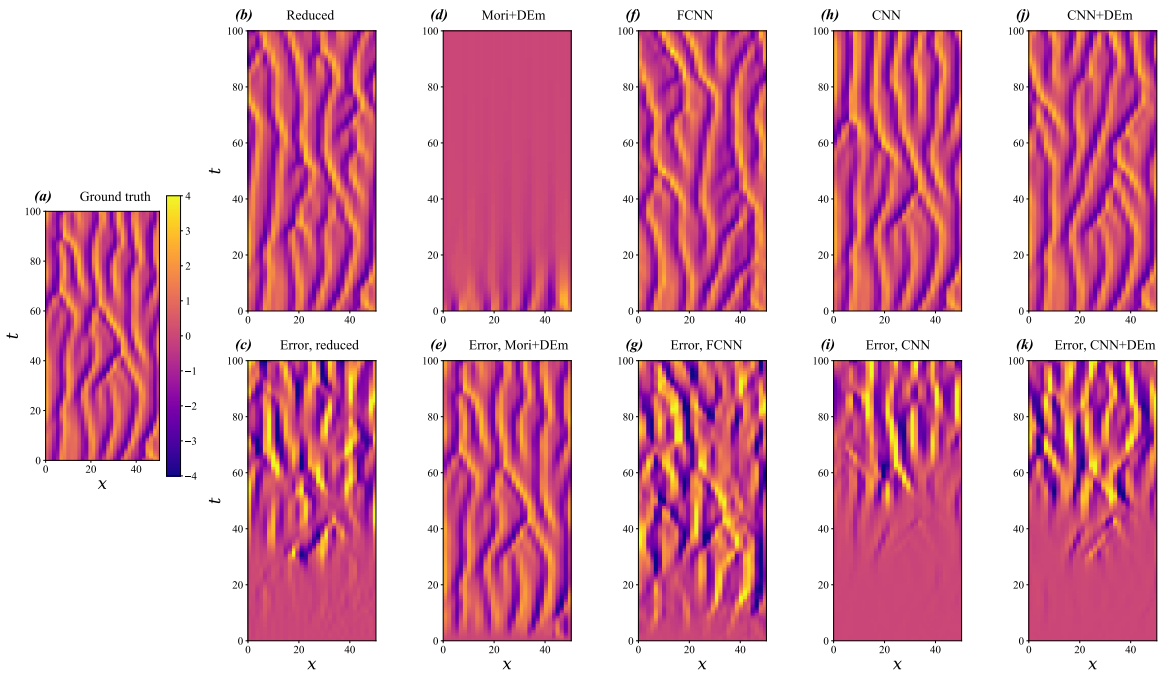

Figure 8. *Regression-based MZ learning of Kuramoto–Sivashinsky equation.* (a) *Ground-truth reduced-order observation.* (b), (d), (f), (h), (j) *Predicted trajectories using reduced-order simulation and four regression-based MZ models, and* (c), (e), (g), (i), (k) *error of the predicted trajectories. The* x- *and* y-*axes of panels* (b)–(k) *are identical to those in panel* (a).

than the $E = 4$ delay-embedded CNN (inset of Figure 9(a)); see the discussion section for a detailed discussion. The FCNN model exhibited a much larger magnitude of error, and a smaller improvement from MZ memory contributions. Finally, as we expected, the linear Mori projection operator with delay embedding had a very poor performance in one-step prediction.

We next focused on predictions at a further horizon. We set the furthest prediction horizon at $t = 100$, which is approximately 8 Lyapunov time. In Figure 9(b), we compare the MSE as a function of the predictive horizon. It was observed that including the MZ memory improved the multistep predictions. Similar to the Lorenz '63 system, at short prediction horizon ($\lesssim$20), models with a nonlinear projection operator performed much better than the linear Mori's model. Among the nonlinear models, the FCNN model had worse predictions than the reduced simulation, and both CNN models outperformed the reduced simulation. Comparing both CNN models, the delay-embedded CNN models performed better than the plain CNN at one-step prediction, as established in Figure 9(a). However, the CNN model overtook the time-embedding CNN model and became the best model—in the metric of the MSE—among all the considered models before the prediction horizon exceeded 55. As illustrated with the Lorenz (1963) model, the MSE is a less meaningful metric for the long-horizon predictions. As a second metric to quantify model performances, we used the Kullback–Leibler divergence from the regression model predicted marginal distribution (u on a single grid) to that of the ground-truth (test trajectory) distribution. Because of the translational symmetry, we accumulated all the measured $u(t, x_{4i}), i = 0 \ldots 31$, at different horizons for

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

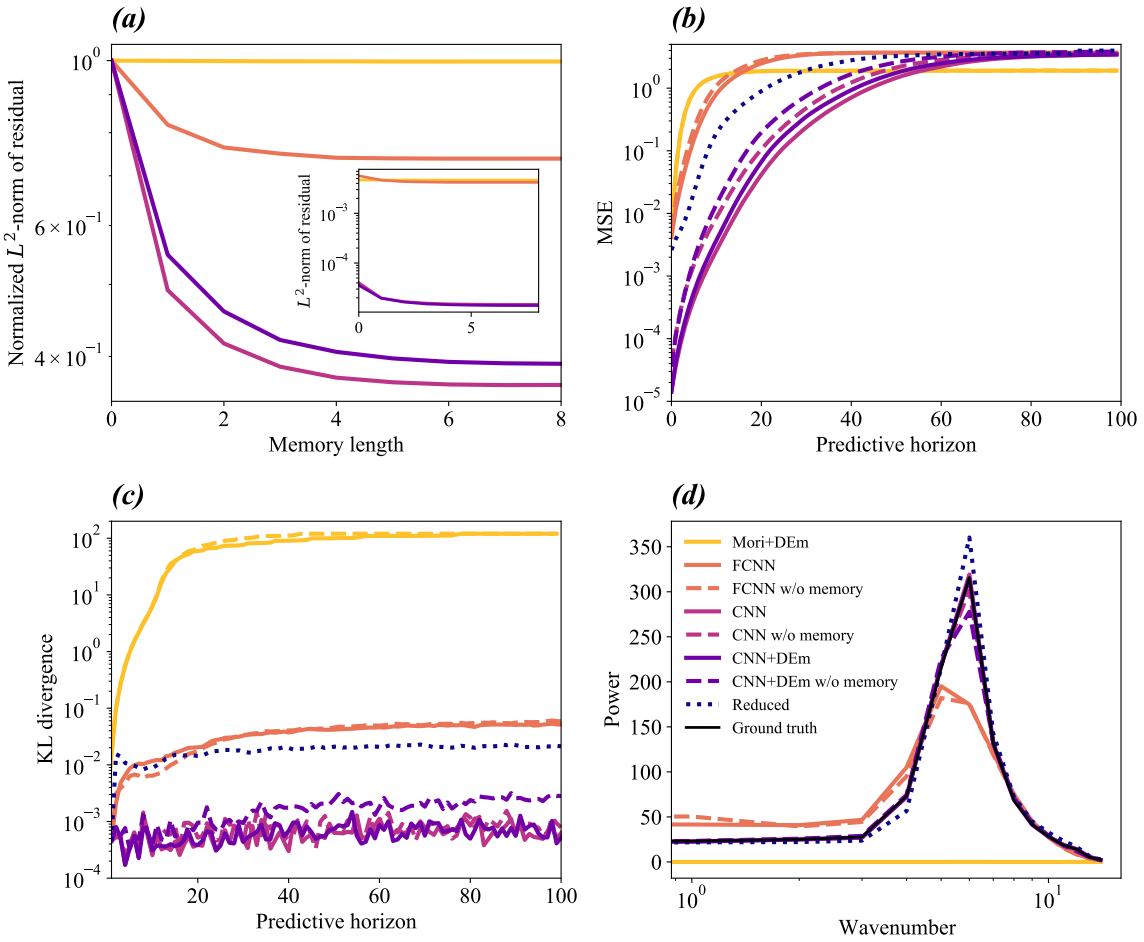

Figure 9. *MZ learning of Kuramoto–Sivashinsky system with regression-based projection operator.* (a) *The MSE of the learned models as a function of memory length.* (b) *The mean squared prediction error over different prediction horizon. The Lyapunov time for chosen domain size* $L = 16\pi$ *is approximately* 12.56. (c) *The Kullback–Leibler divergence of the marginal* $(u(t, x_{4i})\ \forall i = 0 \dots 31)$ *ground-truth distribution from the predicted distribution as a function of prediction horizon.* (d) *The power spectrum of different models at a prediction horizon* 100.

computing the empirical one-dimensional distributions of u. Figure 9(c) shows the computed Kullback–Leibler divergence from the predicted distributions at different prediction horizons. Finally, we compared the power spectra at the prediction horizon 100; Figure 10 also shows the spectrograms of the models with a single sampled initial condition. With these metrics, we concluded that the CNN models, with the MZ memory, performed much better than all other models and the reduced simulations over a long prediction horizon. Again, the presented evidence illustrates the advantage of nonlinear projection at predicting statistical properties of the dynamical system at a longer prediction horizon, despite the fact that samplewise, the predicted trajectories deviate from the ground-truth trajectories.

The long-time marginal distribution of the regression-based models and the reduced simulation is shown in Figure 11. The empirical distributions were computed by accumulating

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

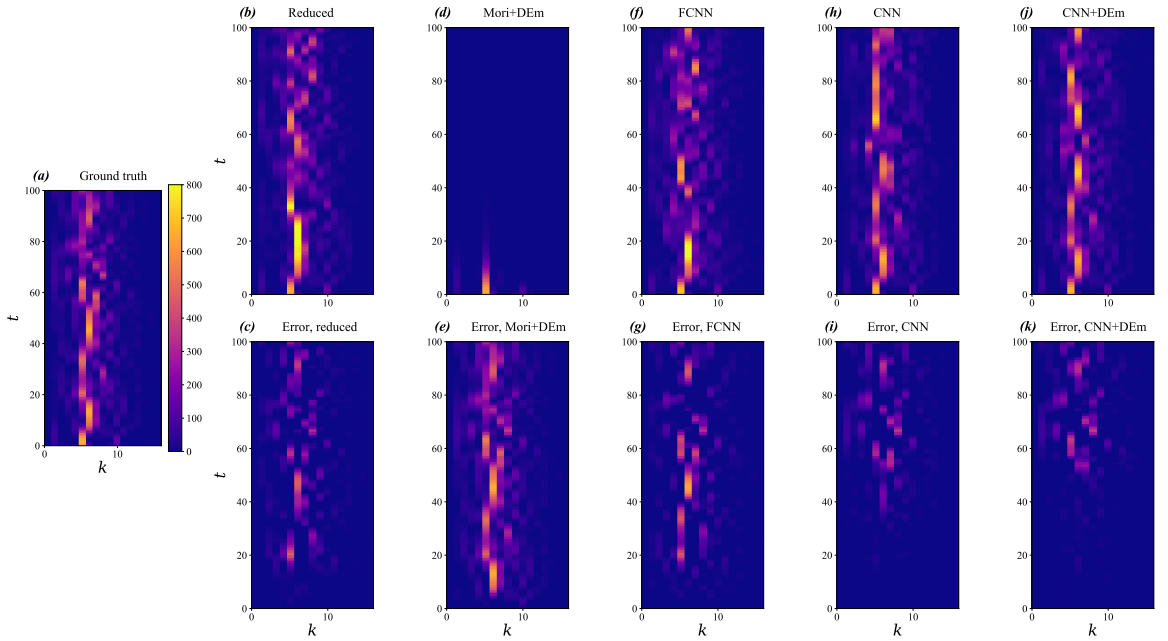

Figure 10. *Projection-based MZ learning of Kuramoto–Sivashinsky system.* (a) *Ground-truth spectrum of the coarse-grained trajectory.* (b), (d), (f), (h), (j) *Predicted spectrum using reduced-order simulation and four regression-based MZ models, and* (c), (e), (g), (i), (k) *error of the predicted spectrum.*

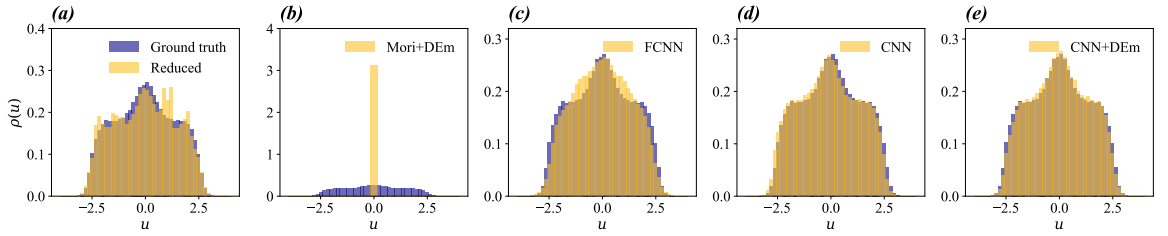

Figure 11. *Projection-based MZ learning of Kuramoto–Sivashinsky system.* (a)–(e) *PDFs of long-time predicted trajectories (horizon* = 10,000*) for five regression-based MZ models.*

measured u on each grid along long predicted trajectories (15,000 steps). The initial conditions of all the models, as well as the reduced simulation, were all set as u_0^{te}. Mori's projection with delay embedding showed almost δ-like distribution at the mean as expected, while the nonlinear models were capable of reproducing the stationary marginal distribution similar to the ground-truth. Among the nonlinear models, the distributions from both CNN models almost perfectly aligned with that of the test trajectory, while the FCNN model missed some finer details around the mean.

5. Discussion and conclusion. The central proposition of this article is to adopt the regression analysis to define the projection operator in the MZ formalism to enable data-driven learning of the operators. Given a set of sampled pairs of dependent and independent variables, a regression analysis identifies an optimal parametric function of the independent

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

variables, such that the residuals are minimized in some norm, prescribed by the noise model. In the context of dynamical systems, the dependent variables are the future observations, and the independent variables are the present ones. When the system is not fully resolved, the future observations cannot always be uniquely determined by the present ones due to missing information of the unresolved degrees of freedom. Consequently, future observations can be multivalued functions of the present ones in underresolved systems. A regression projects such multivalued functions to single-valued functions, constrained by the data. It is a projection operator because an additional regression analysis on the samples drawn from the best-fit model results in the best-fit model, satisfying the definition of a projection operator $\hat{\mathcal{P}}$: $\hat{\mathcal{P}}^2 = \hat{\mathcal{P}}$.

To identify regression analysis as a projection operator is not a novel idea; it is an established concept in statistics. For example, see [9] for linear regression as an orthogonal projection and [36] for regularized regression in terms of the reproducing kernel Hilbert space. The notion of decomposing the samples of the dependent variables into a regressional and an orthogonal component [1] even went back to Kolmogorov in the 1940s [19]. The major conceptional contribution of this manuscript is the proposition *to adopt a regression analysis as the projection operator in the Mori–Zwanzig formalism.* This is a novel concept which, to our best knowledge, does not exist in the current literature. Then, we used the GFD relationship, a key self-consistent condition in the MZ formalism, to recursively extract the memory kernels. Our novel approach enforces the GFD, which is neither enforced nor checked in existing methods [28, 14] motivated by the MZ formalism. In doing so, we are able to directly extract the MZ memory operators computationally, without modeling them with additional assumptions.

We showed that the special choice of adopting linear regression models results in our previously proposed data-driven learning algorithm for the MZ formalism with Mori's projection operator [24]. Thus, the proposed regression-based projection operator is a generalization of our previous proposition. Our proposed procedure to extract the MZ memory kernels is readily applicable to a large set of data-driven learning methods for dynamical systems in the literature because they can be cast into the form of (2.15)–(2.18). For example, approximate Koopman methods with neural network–based learning of the functional bases with ridge and sparsity-induced regularizations [22, 27, 51], and sparse identification of nonlinear dynamics [5, 17]. We remark that these data-driven learning methods often implicitly assume that the system is fully resolved without missing information. Should it be the case and should the regression model be expressive enough, our procedure would lead to a perfect Markov model $\mathbf{\Omega}^{(0)}$ that leaves zero residuals. Our interest, as well as the setup of the MZ formalism, is on partially resolved dynamical systems. In general, when a system is not fully resolved and when there exists interaction between the resolved and unresolved dynamics, MZ memory kernels and orthogonal dynamics naturally emerge.

Regression-based MZ methods can fill in the gap between the existing Mori's and Zwanzig's projection operators, which can be respectively considered as the simplest and the most optimal projection operator [8]. Our proposition significantly broadens the applicability of the MZ formalism, because the complexity of the regression model can be gradationally adjusted between these two extremes. Ranked by the complexity of the regression analysis, we examined and presented the linear projection with [24] and without [44, 52] memory, and various regression techniques such as polynomial regression, spline regression with ridge regularization, and

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

the more modern and expressive neural network architectures such as FCNN and CNN for performing nonlinear projection, with or without memory. With a finite snapshot data set, it may not be computationally feasible to learn the operators with the most optimal Zwanzig's projection operator, because it requires estimating the conditional expectation functions (e.g., $f(X) := \mathbb{E}[Y|X]$). A finite set of data may not be sufficient to cover all possible condition X, let alone the large amount of samples of Y needed for estimating the expectation value. Instead, regression analysis identifies the best fit among a family of functions to approximate the conditional expectation function. When the family of functions is expressive enough to include the conditional expectation function, our regression-based MZ would converge to the Zwanzig's projection operator in the infinite-data limit. As neural networks are universal function approximators [35], we hypothesize that the MZ formalism with the optimal Zwanzig projection operator can be reasonably realized by a neural network–based regression. Indeed, in our numerical experiments, we observed the superiority of those regression models based on neural networks.

Mathematically, the presented method learns the *operators* $\mathbf{\Omega}^{(\ell)}$, which map square integrable functions to square integrable functions, in (2.7). Computationally, this is made possible by using the regressional functions to model the *functions* $\mathbf{\Omega}^{(\ell)}$ in (2.5), given snapshots of the observations $\mathbf{g}(\boldsymbol{\phi}(t))$, where the state is sampled from the induced distribution $\boldsymbol{\phi}(t) = \boldsymbol{\phi}(t; \boldsymbol{\phi}_0)$, $\boldsymbol{\phi}_0 \sim \mu$. It should be noted that (machine) learning of functional operators is not a new idea. For example, Lu et al. [26] recently laid out a general machine learning architecture (DeepONet) for learning nonlinear functional operators, relying on a universal approximation theorem proved in the 1990s [50]. Another example is the approximate Koopman learning methods [44, 45, 52], with which one aims to learn the (approximate) Koopman operator, linearly mapping functions to functions. With these methods, samples of function evaluations must be provided for supervised learning of the operators. Where the functions are evaluated—where the "sensors" are in the language of [26]—requires human inputs. With regard to our aim of learning dynamical systems, analogously, we also need to specify the initial distribution μ. What is different from the general DeepONet [26] is that we can rely on the dynamics to naturally propagate the specified initial distribution to a future time $\boldsymbol{\phi}(t; \boldsymbol{\phi}_0)$, with which the samples of the observations can be generated for supervised learning—in our case, a general regression problem.

We have been framing the above discussions in terms of data-driven learning methods. In fact, our original desire for developing data-driven MZ methods with nonlinear projection operators emerged from *modeling* reduced-order dynamical systems. Devising closure schemes is not only necessary but also critical in these modeling endeavors, especially for multiscale models. Based on our knowledge, there are very few closure schemes which resemble the linear projection operators (i.e., approximate Koopman [44, 52] and Mori [24]), with which one predicts the evolution of a set of observables by linear superposition of the same set of observables. More often, one adopts the nonlinear projection scheme defined in section 3.3 and iteratively applies the learned nonlinear map (i.e., (3.18)) to make multistep predictions. By formally defining a regression (or more generally, data-driven parametrization) as the projection operator in the MZ formalism, Algorithm 3.1 provides a principled way to extract the MZ memory for a wide class of data-driven modeling and systems identification problems. Unless the residuals are all zero, the extracted memory contributions must be considered to

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

account for the error due to the unresolved dynamics. Including the MZ memory in prediction reduces the error in prediction, as we observed in multiple examples provided in section 4.

In this article, we are primarily interested in the quality of multistep predictions of the learned models. We showed that although linear regression models outperformed nonlinear ones at the far horizon, it is only because the linear models predicted the mean of the dynamics. Because linear models fully ignore higher-order statistical properties of the dynamical systems, we are not confident that such "coarse-grained dynamical systems" are the proper way forward for making predictions. Although such a problem for linear projection operators does not exist when a set of observables linearly spanned an invariant Koopman space, it is extremely difficult to identify the invariant Koopman space in practice. It is our perspective that developing methods based on nonlinear projections (see section 3.3) is the way forward for predictive coarse-grained models. We do remark that methods based on linear projection operators are ideal for estimating the Koopman eigenvalues and eigenfunctions, and the globally convex learning problem is easier, provided a set of linearly independent observables. We should also remark that nonlinear regression is not always advantageous in making predictions. As we witnessed in the Lorenz (1963) model, although a learned nonlinear regression model could improve predictive accuracy within a finite horizon, sometimes these nonlinear closure methods lose long-term stability (Figure 6). Such a pathology is model- and horizon-dependent and, interestingly, not observed when we used neural network–based regression. Enforcing numerical stability in the process of learning merits further investigation and may be a fruitful direction in the future.

With the Kuramoto–Sivashinsky model, we also compared the delay embedding and the memory effect of the MZ formalism. Our motivation was to provide clarification on pervasive confusion between these two memory-dependent theories. Such confusion emerged from a false presumption that all memory-dependent theories are equivalent. Here, we elaborate the differences between these two approaches.

Because the memory effect in MZ formalism is due to the interaction between the resolved and unresolved dynamics, the memory kernels and the orthogonal dynamics must satisfy the stringent GFD relation (equation (2.11)), which depends on the choice of the resolved observables and the projection operator. Guided by the GFD (2.11), learning MZ operators can be decomposed into an H smaller regression (optimization) problem in M dimensions. Notably, the MZ memory contribution is always a linear sum of memory kernels evaluated at each past snapshot (equation (2.5)). The formalism excludes coupling between the observations made at different times, for example, $\mathbf{g}_{n_1} \times \mathbf{g}_{n_2}$, with $n_1 \neq n_2$. This is because the projection operator $\mathcal{P}$ is applied at each time by construction. The functional space that $\mathcal{P}$ is projected into is those regressional functions $f(\cdot;\theta)$, which always only depend on M reduced-order observables evaluated at a specific time.

On the contrary, for delay embedding, the domain of the functions is augmented to include E past snapshots of M reduced-order observables. The learning is a global regression (optimization) problem in an $M \times E$-dimensional functional space. The motivation of delay embedding is that the past history may contain useful information for inferring the unresolved dynamics. Because such an augmented functional space $(M \times E)$ is larger than the fixed (M) operational space in the MZ formalism, delay embedding can be considered as a more expressive regression model. Let us denote the past E snapshots of the observables $\mathbf{g}_{0,-1\ldots-E+1}$ by a

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

flattened vector $\mathbf{g}_{0:-E+1}\left(\boldsymbol{\phi}_{-T}\right) \in \mathbb{R}^{ME}$. Formally, delay embedding can be considered as a regression to identify the optimal function $\mathbf{f}_{\boldsymbol{\theta}^*}\left(\mathbf{g}_{0:-E+1}\left(\boldsymbol{\phi}_{-T}\right)\right)$ that best approximates $\mathbf{g}_1\left(\boldsymbol{\phi}_0\right)$. As such, coupled terms could exist, should the family of regressional functions admit, within the delay-embedding paradigm. Hankel-DMD [3] is the special delay-embedding case that the learning is formulated as a linear regression problem, i.e., the regression functions take the form $\mathbf{f}\left(\mathbf{g}_{0:-E+1}\left(\boldsymbol{\phi}_0\right); \boldsymbol{\beta}\right) := \boldsymbol{\beta} \cdot \mathbf{g}_{t-E+1:t}\left(\boldsymbol{\phi}_0\right)$, where $\boldsymbol{\beta}$ is an $(ME) \times (ME)$ matrix (versus in linear regression–based MZ, $\boldsymbol{\beta}^{(\ell)}$ is $M \times M$, $\ell = 0 \ldots H-1$ with the same memory length). Although its structure resembles Mori and Zwanzig's GLE, it is not necessary—and generally not the case—that the best-fit parameters of the Hankel-DMD satisfy the GFD relation (2.11). Notably, an extreme of the delay-embedding technique is the Wiener projection, which has an infinitely long delay embedding. For stationary processes, Wiener projection leads to no MZ memory [23]. A similar notion was also concluded in [14], in which the authors argue that kernel methods could be used to learn the delay embedding with sufficient information that fully resolves the dynamics, resulting in zero MZ memory contribution. Our proposition complements these approaches which attempt to eliminate the MZ memory kernels: given any projection operator, our proposed procedure extracts the MZ memory kernels.

Because the structure of the GLE is contained in the family of delay-embedded regressional functions, the error of the former should have been lower-bounded by that of the latter. However, with the Kuramoto–Sivashinsky equation, we showed that the MZ formalism with $H = 4$ truncated memory contribution outperformed the more expressive delay-embedded CNN with $E = 4$. We hypothesized that it was because the architecture of the delay-embedded CNN was not as expressive as the CNN with MZ memory. As the latter contained four separate CNNs, each of which learned an operator $\boldsymbol{\Omega}^{(\ell)}$, the delay-embedded CNN had a fixed architecture (the past history augmented the channel dimension of the input and not the complexity of the regression model). To examine whether a more expressive delay-embedded CNN could learn better, we deliberately developed delay-embedded CNNs whose number of parameters is identical to H times the number of parameters of CNNs for learning MZ operators. For $H = E = 4$, we still observed that the MZ outperformed the delay-embedding model (1-step MSE of the CNN + MZ: 1.563×10^{-5}; 1-step MSE of the CNN + DEm: 2.038×10^{-5}). For $H = E = 10$, the delay-embedding model performed better in predicting the first step than the MZ did (1-step MSE of the CNN + MZ: 1.478×10^{-5}; 1-step MSE of the CNN + DEm: 1.085×10^{-5}). However, for multistep predictions, MZ outperformed the delay embedding after a few ($\mathcal{O}(10)$) steps; see Figure 12. These observations highlight the practical difficulty of learning the larger delay-embedded models: they require more data, potentially need a longer history, and for nonlinear problems, the training could be trapped in some local minimum. The precise reason why the delay-embedded models are worse in predicting multiple steps into the future merits a more carefully designed investigation.

Furthermore, we demonstrated that MZ formalism and delay embedding are not mutually exclusive. On the Kuramoto–Sivashinsky model, we illustrated a delay-embedded CNN combined with the MZ formalism. In this combined approach, the input of the Markov operator was a flattened observable $\mathbf{g}_{0:-E+1}$, and the input of the ℓth-order ($\ell \geq 1$) memory operator was a vector function of the flattened observable shifted by ℓ time step, $\mathbf{g}_{-\ell:-E-\ell+1}$. We show in Figure 9(a) and (b) that the inclusion of the MZ memory operators can further improve the performance of the delay-embedded CNN.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

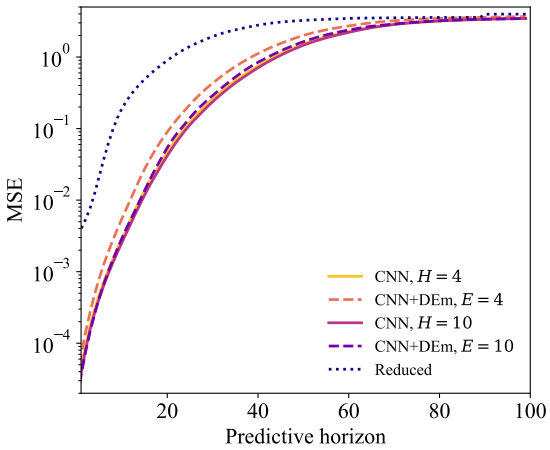

Figure 12. *MZ learning of Kuramoto–Sivashinsky system with various CNN-based regression models. The MSE of the prediction as a function of the prediction horizon. The CNN-based MZ models with different MZ memory lengths ($H = 4$ and $H = 10$) are compared with corresponding delay-embedded models with identical set of embedding lengths ($E = 4$ and $E = 10$). The CNN architectures for this numerical experiment are specially designed such that* (1) *the total number of parameters of both the CNN-based MZ model and the CNN-based delay-embedded model is the same and* (2) *the same information, i.e., the length of the past history, is used for making predictions for both models ($H = E$).*

We view regression-based MZ formalism as a promising direction to enable data-driven learning for partially observed dynamical systems. We have identified a few directions which merit further investigation. Theoretically, we would like to generalize the formalism for stochastic systems, including generic stochastic processes, hidden Markov models, and random dynamical systems. We are working on the application of the neural network–based MZ models on isotropic turbulence data [49]. Another interesting idea we would like to explore is making formal connections to time-series analysis. As recently pointed out, dynamic mode decomposition can be considered as a vector autoregression (VAR-1) model in time-series analysis [4]. It is clear that the Hankel-DMD can be mapped to a VAR-p model in time-series analysis. It was shown that the Wiener projection can be mapped to the nonlinear autoregressive moving average model with exogenous inputs (NARMAX) in time-series analysis, after a rational approximation in the Laplace domain [23]. This existing evidence prompted an interesting research question: what is the time-series analysis method which corresponds to the MZ formalism, which imposes a very stringent GFD condition on the residuals and the memory? Next, we discussed the difference between the memory-embedding techniques such as Hankel-DMD and the MZ formalism. These two formalisms can be considered as two extremes of memory-dependent dynamics: the former considers an extremely large functional space which couples a whole segment of the past trajectory, and the latter restricts its functional space to the regressional functions at an instantaneous time. In modern machine learning for time series, RNN-LSTM [16] and its variants are also equipped with memory kernels, learned by a neural network. The architecture has been claimed to learn MZ memory kernels [28, 15]. We argue that because the critical GFD is neither enforced nor checked, it is unclear that RNN-LSTM is learning MZ memory, or some other form of memory-dependent

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

predictors. It will be an interesting direction to perform a careful analysis, dissecting the RNN-LSTM to identify what type of memory-dependent predictor RNN-LSTM is. Finally, in this manuscript, we did not attempt to model the orthogonal dynamics, which is needed for making accurate predictions. Our test showed that the common choice of white (independent in time) Gaussian noise is unsatisfactory (data not shown). A certain color noise must be needed. This is because the orthogonal dynamics, which encode unresolved dynamics, are also correlated in time. Although our numerical extraction of the orthogonal dynamics provides the data stream for learning, how one can use these extracted data to model orthogonal dynamics in a principled and system-agnostic way remains an open and challenging problem.

REFERENCES

[1] S. N. Afriat, *Regression and Projection*, Springer, Berlin, 1969, pp. 277–301, https://doi.org/10.1007/978-3-642-46198-9_12.

[2] H. Arbabi, M. Korda, and I. Mezić, *A data-driven Koopman model predictive control framework for nonlinear partial differential equations*, in Proceedings of the Conference on Decision and Control, IEEE, 2018, pp. 6409–6414.

[3] H. Arbabi and I. Mezić, *Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the Koopman operator*, SIAM J. Appl. Dyn. Syst., 16 (2017), pp. 2096–2126, https://doi.org/10.1137/17M1125236.

[4] E. 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, 31 (2021), 013108, https://doi.org/10.1063/5.0024890.

[5] S. L. Brunton, J. L. Proctor, and J. N. Kutz, *Discovering governing equations from data by sparse identification of nonlinear dynamical systems*, Proc. Natl. Acad. Sci. USA, 113 (2016), pp. 3932–3937, https://doi.org/10.1073/pnas.1517384113.

[6] A. J. Chorin, O. H. Hald, and R. Kupferman, *Optimal prediction with memory*, Phys. D, 166 (2002), pp. 239–257, https://doi.org/10.1016/S0167-2789(02)00446-3.

[7] A. J. Chorin and F. Lu, *Discrete approach to stochastic parametrization and dimension reduction in nonlinear dynamics*, Proc. Natl. Acad. Sci. USA, 112 (2015), pp. 9804–9809, https://doi.org/10.1073/pnas.1512080112.

[8] E. Darve, J. Solomon, and A. Kia, *Computing generalized Langevin equations and generalized Fokker-Planck equations*, Proc. Natl. Acad. Sci. USA, 106 (2009), pp. 10884–10889, https://doi.org/10.1073/pnas.0902633106.

[9] M. P. Deisenroth, A. A. Faisal, and C. S. Ong, *Mathematics for Machine Learning*, Cambridge University Press, Cambridge, UK, 2020, https://doi.org/10.1017/9781108679930.

[10] P. Durbin, *Some recent developments in turbulence closure modeling*, Annu. Rev. Fluid Mech., 50 (2018), pp. 77–103.

[11] R. A. Edson, J. E. Bunder, T. W. Mattner, and A. J. Roberts, *Lyapunov exponents of the Kuramoto-Sivashinsky PDE*, ANZIAM J., 61 (2019), pp. 270–285, https://doi.org/10.1017/S1446181119000105.

[12] U. Fasel, J. N. Kutz, B. W. Brunton, and S. L. Brunton, *Ensemble-SINDy: Robust sparse model discovery in the low-data, high-noise limit, with active learning and control*, Proc. R. Soc. A, 478 (2022), 20210904.

[13] G. Gerlich, *Die verallgemeinerte Liouville-Gleichung*, Physica, 69 (1973), pp. 458–466, https://doi.org/10.1016/0031-8914(73)90083-9.

[14] F. Gilani, D. Giannakis, and J. Harlim, *Kernel-based prediction of non-markovian time series*, Phys. D, 418 (2021), 132829.

[15] J. Harlim, S. W. Jiang, S. Liang, and H. Yang, *Machine learning for prediction with missing dynamics*, J. Comput. Phys., 428 (2021), 109922.

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[16] S. Hochreiter and J. Schmidhuber, *Long short-term memory*, Neural Comput., 9 (1997), pp. 1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735.
[17] K. Kaheman, J. N. Kutz, and S. L. Brunton, *SINDy-PI: A robust algorithm for parallel implicit sparse identification of nonlinear dynamics*, Proc. A, 476 (2020), 20200279, https://doi.org/10.1098/rspa.2020.0279.
[18] A.-K. Kassam and L. N. Trefethen, *Fourth-order time-stepping for stiff PDEs*, SIAM J. Sci. Comput., 26 (2005), pp. 1214–1233, https://doi.org/10.1137/S1064827502410633.
[19] A. N. Kolmogorov, *On the proof of the method of least squares*, Uspekhi Mat. Nauk, 1 (1946), pp. 57–70.
[20] B. O. Koopman and J. v. Neumann, *Dynamical systems of continuous spectra*, Proc. Natl. Acad. Sci. USA, 18 (1932), pp. 255–263, https://doi.org/10.1073/pnas.18.3.255.
[21] Y. Kuramoto, *Diffusion-induced chaos in reaction systems*, Progr. Theoret. Phys. Suppl., 64 (1978), pp. 346–367, https://doi.org/10.1143/PTPS.64.346.
[22] Q. Li, F. Dietrich, E. M. Bollt, and I. G. Kevrekidis, *Extended dynamic mode decomposition with dictionary learning: A data-driven adaptive spectral decomposition of the Koopman operator*, Chaos, 27 (2017), 103111, https://doi.org/10.1063/1.4993854.
[23] K. K. Lin and F. Lu, *Data-driven model reduction, Wiener projections, and the Koopman-Mori-Zwanzig formalism*, J. Comput. Phys., 424 (2021), 109864, https://doi.org/10.1016/j.jcp.2020.109864.
[24] Y. T. Lin, Y. Tian, D. Livescu, and M. Anghel, *Data-driven learning for the Mori-Zwanzig formalism: A generalization of the Koopman learning framework*, SIAM J. Appl. Dyn. Syst., 20 (2021), pp. 2558–2601, https://doi.org/10.1137/21M1401759.
[25] E. N. Lorenz, *Deterministic nonperiodic flow*, J. Atmos. Sci., 20 (1963), pp. 130–141, https://doi.org/10.1175/1520-0469(1963)020⟨0130:DNF⟩2.0.CO;2.
[26] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis, *Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators*, Nat. Mach. Intell., 3 (2021), pp. 218–229, https://doi.org/10.1038/s42256-021-00302-5.
[27] B. Lusch, J. N. Kutz, and S. L. Brunton, *Deep learning for universal linear embeddings of nonlinear dynamics*, Nat. Commun., 9 (2018), https://doi.org/10.1038/s41467-018-07210-0.
[28] C. Ma, J. Wang, and W. E, *Model reduction with memory and the machine learning of dynamical systems*, Commun. Comput. Phys., 25 (2018), pp. 947–962.
[29] S. Maeyama and T.-H. Watanabe, *Extracting and modeling the effects of small-scale fluctuations on large-scale fluctuations by Mori-Zwanzig projection operator method*, J. Phys. Soc. Jpn., 89 (2020), 024401.
[30] A. J. Majda and X. Wang, *Nonlinear Dynamics and Statistical Theories for Basic Geophysical Flows*, Cambridge University Press, Cambridge, UK, 2006.
[31] H. Meyer, P. Pelagejcev, and T. Schilling, *Non-Markovian out-of-equilibrium dynamics: A general numerical procedure to construct time-dependent memory kernels for coarse-grained observables*, Europhys. Lett. EPL, 128 (2020), 40001.
[32] H. Meyer, S. Wolf, G. Stock, and T. Schilling, *A numerical procedure to evaluate memory effects in non-equilibrium coarse-grained models*, Adv. Theory Simul., 4 (2021), 2000197.
[33] H. Mori, *Transport, collective motion, and Brownian motion*, Prog. Theor. Phys., 33 (1965), pp. 423–455.
[34] H. Mori and M. Okamura, *Dynamic structures of the time correlation functions of chaotic nonequilibrium fluctuations*, Phys. Rev. E, 76 (2007), 061104.
[35] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*, MIT Press, Cambridge, MA, 2012.
[36] A. Nosedal-Sanchez, C. B. Storlie, T. C. Lee, and R. Christensen, *Reproducing kernel Hilbert spaces for penalized regression: A tutorial*, Amer. Statist., 66 (2012), pp. 50–60.
[37] M. Okamura, *Validity of the essential assumption in a projection operator method*, Phys. Rev. E, 74 (2006), 046210.
[38] E. J. Parish and K. Duraisamy, *A dynamic subgrid scale model for large Eddy simulations based on the Mori-Zwanzig formalism*, J. Comput. Phys., 349 (2017), pp. 154–175, https://doi.org/10.1016/j.jcp.2017.07.053.
[39] E. J. Parish and K. Duraisamy, *Non-Markovian closure models for large eddy simulations using the Mori-Zwanzig formalism*, Phys. Rev. Fluids, 2 (2017), 14604, https://doi.org/10.1103/PhysRevFluids.2.014604.

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

[40] J. Price, B. Meuris, M. Shapiro, and P. Stinis, *Optimal renormalization of multiscale systems*, Proc. Natl. Acad. Sci. USA, 118 (2021), e2102266118, https://doi.org/10.1073/pnas.2102266118.
[41] J. Price and P. Stinis, *Renormalized reduced order models with memory for long time prediction*, Multiscale Model. Simul., 17 (2019), pp. 68–91, https://doi.org/10.1137/17M1151389.
[42] E. Qian, B. Kramer, B. Peherstorfer, and K. Willcox, *Lift & learn: Physics-informed machine learning for large-scale nonlinear dynamical systems*, Phys. D, 406 (2020), 132401, https://doi.org/10.1016/j.physd.2020.132401.
[43] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson, *Spectral analysis of nonlinear flows*, J. Fluid Mech., 641 (2009), pp. 115–127, https://doi.org/10.1017/S0022112009992059.
[44] P. J. Schmid, *Dynamic mode decomposition of numerical and experimental data*, J. Fluid Mech., 656 (2010), pp. 5–28, https://doi.org/10.1017/S0022112010001217.
[45] P. J. Schmid, L. Li, M. P. Juniper, and O. Pust, *Applications of the dynamic mode decomposition*, Theor. Comput. Fluid Dyn., 25 (2011), pp. 249–259, https://doi.org/10.1007/s00162-010-0203-9.
[46] G. Sivashinsky, *Nonlinear analysis of hydrodynamic instability in laminar flames—*I. *Derivation of basic equations*, Acta Astronaut., 4 (1977), pp. 1177–1206, https://doi.org/10.1016/0094-5765(77)90096-0.
[47] G. I. Sivashinsky, *On flame propagation under conditions of stoichiometry*, SIAM J. Appl. Math., 39 (1980), pp. 67–82, https://doi.org/10.1137/0139007.
[48] F. Takens, *Detecting strange attractors in turbulence*, in Dynamical Systems and Turbulence, D. Rand and L.-S. Young, eds., Springer, Berlin, 1981, pp. 366–381.
[49] Y. Tian, Y. T. Lin, M. Anghel, and D. Livescu, *Data-driven learning of Mori-Zwanzig operators for isotropic turbulence*, Phys. Fluids, 33 (2021), 125118, https://doi.org/10.1063/5.0070548.
[50] T. Chen and H. Chen, *Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems*, IEEE Trans. Neural Netw., 6 (1995), pp. 911–917, https://doi.org/10.1109/72.392253.
[51] C. Wehmeyer and F. Noé, *Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics*, J. Chem. Phys., 148 (2018), 241703, https://doi.org/10.1063/1.5011399.
[52] M. O. Williams, C. W. Rowley, and I. G. Kevrekidis, *A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition*, J. Nonlinear Sci., 25 (2015), pp. 1307–1346, https://doi.org/10.1007/s00332-015-9258-5.
[53] E. Yeung, S. Kundu, and N. Hodas, *Learning deep neural network representations for koopman operators of nonlinear dynamical systems*, in Proceedings of the American Control Conference, IEEE, 2019, pp. 4832–4839.
[54] R. Zwanzig, *Nonlinear generalized Langevin equations*, J. Stat. Phys., 9 (1973), pp. 215–220.
[55] R. Zwanzig, *Nonequilibrium Statistical Mechanics*, Oxford University Press, New York, 2001.

Downloaded 10/13/23 to 192.12.184.6 . Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/terms-privacy

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.