2024 American Control Conference (ACC)
July 8-12, 2024. Toronto, Canada

Structural Risk Minimization for Learning Nonlinear Dynamics

Charis Stamouli, Evangelos Chatzipantazis, George J. Pappas

Abstract— **Recent advances in learning or identification of nonlinear dynamics focus on learning a suitable model within a pre-specified model class. However, a key difficulty that remains is the choice of the model class from which the dynamics will be learned. The fundamental challenge is trading the richness of the model class with the learnability within the model class. Toward addressing the so-called model selection problem, we introduce a novel notion of Structural Risk Minimization (SRM) for learning nonlinear dynamics. Inspired by classical SRM for classification, we minimize a bound on the true prediction error over hierarchies of model classes. The class selected by our SRM scheme is shown to achieve a nearly optimal learning guarantee among all model classes contained in the hierarchy. Employing the proposed scheme along with computable model class complexity bounds, we derive explicit SRM schemes for learning nonlinear dynamics under hierarchies of: i) norm-constrained Reproducing Kernel Hilbert Spaces, and ii) norm-constrained Neural Network classes. We empirically show that even though too loose to be used as absolute estimates of the true prediction error, our SRM bounds are able to track its relative behavior across different model classes of the hierarchy.**

I. Introduction

Modeling nonlinear dynamics to predict the future is of critical importance in a wide range of applications including e.g., fluid dynamics [1], precision medicine [2], and economics [3]. Conventionally, dynamical models are derived from first principles and data are used to estimate their parameters. However, physical systems are often governed by differential equations that are difficult to simulate numerically. Beyond that, mathematical models are not available for many real-world systems. Such systems have motivated the development of learning or identification of nonlinear dynamics, where system models are estimated entirely from data.

The majority of works on identification of nonlinear dynamics [4]–[14] focus on learning a suitable model within a pre-defined model class. The choice of the model class from which the dynamics will be learned is critical to the performance of the learned model. On one hand, a rich enough model class could contain the true dynamics model. On the other hand, learning in such a complex model class could be a very difficult task. Therefore, there is a fundamental trade-off in learning determined by the complexity of the selected model class. The model selection problem consists of choosing the model class that balances this trade-off.

To address the model selection problem, the authors in [15] propose a method for sparse identification of nonlinear dynamics. Their algorithm employs sparse regression and Pareto analysis to learn a parsimonious model from a combinatorial class of candidate models. In the extended version [16], the most informative candidate models are ranked via appropriate information criteria. In both [15] and [16] the only assumption about the dynamics is that they can be sparsely represented. The authors in [17] guarantee this assumption by learning an effective coordinate transformation to a suitable low-dimensional space. Exploiting the Koopman operator, recent works in [18], [19] extend the scheme of [15] to classes of systems with continuous eigenvalue spectra.

In machine learning, a more foundational model selection approach is provided by Structural Risk Minimization (SRM) [20]–[26]. SRM consists of: i) picking a hierarchy of model classes (see Figure 1), and ii) minimizing a class-dependent bound on the true prediction error over the entire hierarchy. In classical machine learning, SRM with strong learning guarantees has been proposed only for classification tasks, where data have binary labels. Particularly, in [20], [21] it is proved that the class corresponding to the respective SRM classifier attains a nearly optimal learning trade-off. In system identification, SRM has been employed for learning switched autoregressive systems with external inputs, under an unknown number of modes [25]. In time series prediction, a general SRM scheme for scalar-valued stationary mixing processes has been developed [26].

In this paper, we introduce a novel notion of SRM for learning nonlinear dynamics. Our setting is different from the one in [26] as it involves vector-valued time series generated by nonlinear dynamical systems that are generally non-stationary and non-mixing. Given a hierarchy of model classes, our method estimates the true dynamics model by minimizing a bound on the true prediction error over the entire hierarchy. The bound is

The authors are with the Grasp Lab, University of Pennsylvania, Philadelphia, PA 19104, USA. Emails: {stamouli,vaghat,pappasg}@seas.upenn.edu.

This work was supported by NSF award EnCORE-2217062.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:27:49 UTC from IEEE Xplore. Restrictions apply.

determined by the prediction error on the dataset and a penalty that captures the complexity of each class. The contributions of our work are the following:

i) We propose the first SRM scheme for learning deterministic nonlinear dynamics with random initial condition, under arbitrary hierarchies of model classes.
ii) We rigorously prove that the class selected by our SRM scheme achieves a nearly optimal learning guarantee among all model classes of the hierarchy.
iii) We design explicit SRM schemes for our setting of nonlinear dynamics, under hierarchies of norm-constrained: a) Reproducing Kernel Hilbert Spaces (RKHSs), and b) Neural Network (NN) classes. This is achieved by combining our scheme with respective computable class complexity bounds.
iv) We empirically show that even though too loose to be used as absolute estimates of the true prediction error, our SRM bounds are able to track its relative behavior across different model classes of the hierarchy.

All proofs can be found in [27, Appendix B].

II. Classical SRM in Machine Learning

Before introducing SRM for our setting of nonlinear dynamical systems (see Section III), we first review classical SRM for the setting of classification [28], [29].

The classification setting is characterized by an instance space $\mathcal{X} \subseteq \mathbb{R}^n$, the label space $\mathcal{Y} := \{-1, +1\}$, an unknown probability distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, and an i.i.d. sample $S := \{(x_i, y_i)\}_{i=1}^N \sim \mathcal{D}^N$. Given the dataset S, the goal is to learn a classifier $h_S : \mathcal{X} \to \mathcal{Y}$ that minimizes the $0-1$ true error (or risk), given by:

$$\mathrm{er}_{\mathcal{D}}^{0-1}[h] = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathbb{1}(h(x) \neq y)], \tag{1}$$

for each $h : \mathcal{X} \to \mathcal{Y}$, where $\mathbb{1}(\cdot)$ denotes the indicator function. However, since the distribution $\mathcal{D}$ is unknown, the true error (1) is not available to the learner.

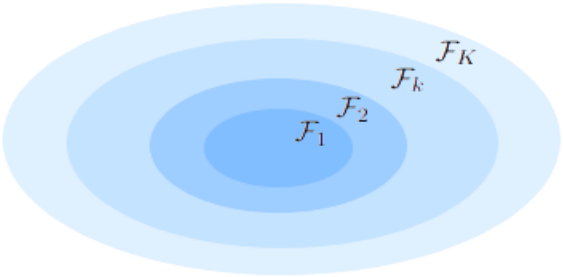


Fig. 1. Illustration of a rich model class $\mathcal{F} = \bigcup_{k=1}^K \mathcal{F}_k$ hierarchized into K nested subclasses $\mathcal{F}_k$. For example, the classes $\mathcal{F}_k$ could correspond to neural networks of increasing width.

Empirical Risk Minimization (ERM) circumvents this issue by replacing the true error (1) with the corresponding training error:

$$\widehat{\mathrm{er}}_S^{0-1}[h] := \frac{1}{N}\sum_{i=1}^N \mathbb{1}(h(x_i) \neq y_i), \tag{2}$$

which depends only on the dataset S. A key challenge that remains in ERM is model selection, that is, the problem of choosing the function class over which the training error (2) will be minimized. On one hand, a very rich function class is more likely to contain the ideal classifier $h_*(\cdot)$ with the minimum achievable risk. On the other hand, learning in such a complex function class could be a very difficult task. Consequently, there is a trade-off in learning determined by the complexity of the selected function class. To formalize this trade-off, let $\mathcal{H}$ denote a class of functions from $\mathcal{X}$ to $\mathcal{Y}$ and $h_S^{\mathrm{ERM}}(\cdot)$ the classifier returned by ERM on $\mathcal{H}$. We can decompose the risk of $h_S^{\mathrm{ERM}}(\cdot)$ as $\mathrm{er}_{\mathcal{D}}^{0-1}[h_S^{\mathrm{ERM}}] = \mathrm{er}_{\mathrm{app}} + \mathrm{er}_{\mathrm{est}}$, where $\mathrm{er}_{\mathrm{app}} := \min_{h\in\mathcal{H}} \mathrm{er}_{\mathcal{D}}^{0-1}[h]$ is the approximation error and $\mathrm{er}_{\mathrm{est}} := \mathrm{er}_{\mathcal{D}}^{0-1}[h_S^{\mathrm{ERM}}] - \mathrm{er}_{\mathrm{app}}$ is the estimation error. The approximation error is the minimum risk achievable by a classifier in $\mathcal{H}$ and depends on the richness of $\mathcal{H}$. The estimation error measures how well the best classifier of $\mathcal{H}$ is approximated by the ERM classifier $h_S^{\mathrm{ERM}}(\cdot)$ and depends on the dataset S. Notice that the choice of $\mathcal{H}$ is subject to a fundamental trade-off between approximation and estimation errors: in particular, as the complexity of $\mathcal{H}$ increases, the approximation error tends to decrease at the price of a larger estimation error. ERM overlooks this trade-off, which critically affects its practical performance.

SRM addresses the model selection problem, by replacing the true error (1) with a certain upper bound on it that allows regulating the approximation-estimation error trade-off. This is because this bound is not determined only by the training error but also by some class complexity measure, such as Rademacher complexity. The Rademacher complexity captures the richness of a function class by measuring its ability to fit symmetric random noise. Herein, we will focus on the empirical Rademacher complexity $\widehat{\mathcal{R}}_S(\cdot)$ (see [27, Appendix A]), which depends on the sample S. In the theorem below, we present a Rademacher-based SRM scheme and a corresponding learning guarantee for classification. Note that the theorem below is a modified version of [28, Theorem 4.2] for finite hierarchies of function classes.

Theorem 1 (SRM for Classification)**.** *Fix a failure probability $\delta \in (0,1)$. Let $\mathcal{H}$ be a class of functions from $\mathcal{X}$ to $\mathcal{Y}$ that can be written as $\mathcal{H} = \bigcup_{k=1}^K \mathcal{H}_k$ with $\mathcal{H}_k \subset \mathcal{H}_{k+1}$, $\forall k$. Then, for each k, with probability at*

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:27:49 UTC from IEEE Xplore. Restrictions apply.

least $1-\delta$ *over the draw of* $S \sim \mathcal{D}^N$*,* $\mathcal{H}_k$ *satisfies:*

$$\forall h \in \mathcal{H}_k : \left| \mathrm{er}_{\mathcal{D}}^{0-1}[h] - \widehat{\mathrm{er}}_S^{0-1}[h] \right| \leq \epsilon_k(S,\delta), \quad (3)$$

where $\epsilon_k(S,\delta) = \widehat{\mathcal{R}}_S(\mathcal{H}_k) + 3\sqrt{\log(4/\delta)/(2N)}$*. Moreover, if the SRM classifier is given by:*

$$h_S^{\mathrm{SRM}} = \underset{1\leq k\leq K, h\in\mathcal{H}_k}{\arg\min} \left(\widehat{\mathrm{er}}_S^{0-1}[h] + \widehat{\mathcal{R}}_S(\mathcal{H}_k) \right), \quad (4)$$

then, with probability at least $1-\delta$ *over the draw of* $S \sim \mathcal{D}^N$*, the following bound holds:*

$$\mathrm{er}_{\mathcal{D}}^{0-1}[h_S^{\mathrm{SRM}}] \leq \min_{h\in\mathcal{H}} \left(\mathrm{er}_{\mathcal{D}}^{0-1}[h] + 2\epsilon_{k(h)} \left(S, \frac{2\delta}{K+1} \right) \right), \quad (5)$$

where $k(h) = \min\{k \,|\, h \in \mathcal{H}_k\}$.

Notice in (4) that in contrast to ERM, SRM does not minimize just the training error, but the sum of the training error and the penalty $\widehat{\mathcal{R}}_S(\mathcal{H}_k)$. Therefore, SRM trades some of its bias toward low training error with a bias toward classes $\mathcal{H}_k$ for which $\widehat{\mathcal{R}}_S(\mathcal{H}_k)$ is small, for the sake of a smaller estimation error. The SRM guarantee (5) ensures that the selected class achieves in fact a nearly optimal balance between approximation and estimation errors (see [28] for details).

III. SRM for Learning Nonlinear Dynamics

In the previous section, we summarized classical SRM for classification. We are now ready to introduce SRM for our setting of nonlinear dynamical systems. In Subsection III-A, we formulate the problem of SRM for deterministic nonlinear dynamics with random initial condition. In Subsection III-B, we provide the first SRM scheme in the literature for this setting, along with a corresponding learning guarantee.

A. SRM Problem for Nonlinear Dynamics

Consider a discrete-time nonlinear dynamical system of the form:

$$x_{t+1} = f_*(x_t), \quad (6)$$

where $x_t \in \mathbb{R}^n$ denotes the state at time t and $f_* : \mathbb{R}^n \to \mathbb{R}^n$ the dynamics function, which is assumed to be unknown. The initial state x_0 is modeled as a random variable with an unknown probability distribution $\mathcal{D}$. Let $T \geq 1$ be the time horizon over which trajectories generated by system (6) are considered (e.g., T could be the system operation time). Moreover, let $x_{0:T}(\xi) = \left[x_0(\xi), \ldots, x_T(\xi)\right]^{\mathsf{T}}$, where $x_t(\xi)$ denotes the state of system (6) at time t, for some initial condition $\xi \sim \mathcal{D}$.

Suppose we are given a set $S := \{x_{0:T}(\xi_i)\}_{i=1}^N$ of N trajectories of system (6) corresponding to an i.i.d. sample of initial conditions $S_0 := \{\xi_i\}_{i=1}^N \sim \mathcal{D}^N$. Note the implicit dependence of S on S_0. Given the dataset S, the objective is to learn a predictor $f_S : \mathbb{R}^n \to \mathbb{R}^n$ such that for each $\xi \sim \mathcal{D}$, we have $f_S(x_t(\xi)) \approx f_*(x_t(\xi))$, for all $t = 0, \ldots, T-1$, that is, $f_S(\cdot)$ should be able to approximate the next state at each time step t. Formally, given the dataset S and a loss function $\ell : \mathbb{R}^n \to \mathbb{R}_+$, we seek a predictor $f_S : \mathbb{R}^n \to \mathbb{R}^n$ that minimizes the corresponding true error (or risk), given by:

$$\mathrm{er}_{\mathcal{D}}[f] = \mathbb{E}_{\xi\sim\mathcal{D}} \left[\frac{1}{T} \sum_{t=0}^{T-1} \ell\left(f(x_t(\xi)) - x_{t+1}(\xi) \right) \right], \quad (7)$$

for each $f : \mathbb{R}^n \to \mathbb{R}^n$.

Assumption 1. *The loss function* $\ell(\cdot)$ *is locally Lipschitz continuous and satisfies* $\ell(0) = 0$*.*

The above assumption is satisfied by all standard regression loss functions (e.g., L_p loss for each $p \geq 1$, Huber loss). Notice that the true error (7) cannot be minimized directly as it depends on the unknown distribution $\mathcal{D}$. As in the case of classification (see Section II), we can tackle this challenge by employing ERM, which replaces the true error (7) with the corresponding training error:

$$\widehat{\mathrm{er}}_S[f] := \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=0}^{T-1} \ell\left(f(x_t(\xi_i)) - x_{t+1}(\xi_i) \right). \quad (8)$$

Assumption 2. *There exists a known bound* $B > 0$ *such that* $\|x_t(\xi)\|_2 \leq B$*, for all* $t = 0, \ldots, T$ *and* $\xi \sim \mathcal{D}$*.*

Remark 1. *The above assumption implies that the predicted states should lie in the Euclidean ball of radius* B*. To guarantee that, we employ the method of clipping, which is commonly used in machine learning to obtain bounded predictors by searching over unbounded model classes [28]. In particular, for each predictor* $f : \mathbb{R}^n \to \mathbb{R}^n$*, we define its clipped version* $\bar{f}(\cdot)$ *as:*

$$\bar{f}(x) = \begin{cases} \frac{B}{\|f(x)\|_2} \cdot f(x), & \text{if } \|f(x)\|_2 > B \\ f(x), & \text{else} \end{cases}, \quad (9)$$

for all $x \in \mathbb{R}^n$*. Note that in practice the constant* B *could be a rough bound that we know from physics or have obtained from observations.*

In classical nonlinear system identification, ERM is employed on some function class $\mathcal{F}$ which is assumed to contain the true dynamics model [30]. This assumption, though, is typically unrealistic in practice, unless the selected function class $\mathcal{F}$ is very rich. However, learning in a highly complex function class could be a difficult or even intractable task. Hence, there is a trade-off in learning regulated via the complexity of $\mathcal{F}$. In Section II, we formalized this trade-off by introducing the notions of the approximation and estimation errors. Particularly,

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:27:49 UTC from IEEE Xplore. Restrictions apply.

recall that as the complexity of the class increases, the approximation error decreases at the price of a larger estimation error. But then, how should $\mathcal{F}$ be chosen?

In this paper, we address the above question by formulating SRM for learning nonlinear dynamics. More specifically, our goal is to design a learning algorithm that estimates $f_*(\cdot)$ with a predictor that guarantees a nearly optimal approximation-estimation error trade-off.

Problem 1. *Let $\mathcal{F}$ be a class of functions from $\mathbb{R}^n$ to $\mathbb{R}^n$ that can be written as $\mathcal{F} = \bigcup_{k=1}^{K} \mathcal{F}_k$ with $\mathcal{F}_k \subset \mathcal{F}_{k+1}$, $\forall k$. Moreover, let $L > 0$ be the Lipschitz constant of $\ell(\cdot)$ on the set $Z := \{x \in \mathbb{R}^n \mid \|x\|_2 \leq 2B\}$. Given a failure probability $\delta \in (0,1)$, for each k:*

i) Derive a bound $\epsilon_k(S,\delta)$ such that with probability at least $1-\delta$ over the draw of $S_0 \sim \mathcal{D}^N$:

$$\forall f \in \mathcal{F}_k : \left|\mathrm{er}_{\mathcal{D}}[\bar{f}] - \widehat{\mathrm{er}}_S[\bar{f}]\right| \leq \epsilon_k(S,\delta). \qquad (10)$$

ii) Design a penalty $r_k(S)$ such that with probability at least $1-\delta$ over the draw of $S_0 \sim \mathcal{D}^N$, the SRM predictor:

$$f_S^{\mathrm{SRM}} := \underset{1\leq k\leq K, f\in\mathcal{F}_k}{\arg\min} \left(\widehat{\mathrm{er}}_S[\bar{f}] + r_k(S)\right) \qquad (11)$$

satisfies:

$$\mathrm{er}_{\mathcal{D}}[\bar{f}_S^{\mathrm{SRM}}] \leq \min_{f\in\mathcal{F}} \left(\mathrm{er}_{\mathcal{D}}[\bar{f}] + 2\epsilon_{k(f)}\Big(S, \frac{2\delta}{K+1}\Big)\right), \qquad (12)$$

where $k(f) = \min\{k \mid f \in \mathcal{F}_k\}$.

The bounds $\epsilon_k(S,\delta)$ and the penalties $r_k(S)$ may also depend on the parameters n, N, T, B, L.

Remark 2. *For clarity of presentation, in Problem 1 we assumed that the classes $\mathcal{F}_k$ are nested, as dictated by the condition $\mathcal{F}_k \subset \mathcal{F}_{k+1}$, $\forall k$. However, note that our results directly extend to arbitrary decompositions of $\mathcal{F}$.*

B. SRM Method for Nonlinear Dynamics

In this subsection, we introduce a general SRM scheme for nonlinear dynamics under the setting described in Subsection III-A. Similarly to classical SRM (see Theorem 1), for each k, we provide a Rademacher-based penalty $r_k(S)$ for the SRM predictor (11). The main differences between our setting and that of classical SRM are the following:

i) In our setting, the function that we wish to estimate is *vector-valued*, whereas in classical SRM, it is *scalar-valued*. Particularly, note that $f_*(\cdot)$ takes values in $\mathbb{R}^n$, whereas $h_*(\cdot)$ takes values in $\{-1,+1\}$.

ii) Our dataset consists of N *i.i.d. sequences of dependent datapoints* (trajectories). In classical SRM, the dataset consists of N *i.i.d. datapoints*.

iii) The risk (7) employed in our setting is defined as the expected *average ℓ-loss over trajectories of length T*, whereas the risk (1) employed in classical SRM is simply defined as the expected $0-1$ *loss on a single datapoint*.

Consequently, our framework provides an extension of classical SRM to vector-valued predictors, i.i.d. sequences of data, and a large family of losses, suitably designed for the setting of nonlinear dynamics. Our setting is also different from the one in [26], where general *scalar-valued* time series are considered under extra *stationarity* and *mixing* assumptions. Particularly, our setting involves vector-valued time series, generated by nonlinear dynamics of the form (6), which are generally non-stationary and non-mixing. In the theorem below, we present our Rademacher-based SRM scheme.

Theorem 2 (SRM for Nonlinear Dynamics)**.** *Fix a failure probability $\delta \in (0,1)$. Let $\mathcal{F}$ be a class of functions from $\mathbb{R}^n$ to $\mathbb{R}^n$ that can be written as $\mathcal{F} = \bigcup_{k=1}^{K} \mathcal{F}_k$ with $\mathcal{F}_k \subset \mathcal{F}_{k+1}$, $\forall k$. Moreover, let the SRM predictor $f_S^{\mathrm{SRM}}(\cdot)$ be as in* (11) *with:*

$$r_k(S) = \frac{2\sqrt{2}L}{T} \sum_{t=0}^{T-1} \sum_{j=1}^{n} \widehat{\mathcal{R}}_{S_t}(\bar{\mathcal{F}}_{k,j}), \qquad (13)$$

where $\bar{\mathcal{F}}_{k,j} = \left\{j\text{-th component of } \bar{f}(\cdot) \mid f \in \mathcal{F}_k\right\}$ and $S_t = \{x_t(\xi_i)\}_{i=1}^{N}$. Then, setting $\epsilon_k(S,\delta) = r_k(S) + 6LB\sqrt{\log(4/\delta)/(2N)}$, each of (10) *and* (12) *holds with probability at least $1-\delta$ over the draw of $S_0 \sim \mathcal{D}^N$.*

Note that the definition of the empirical Rademacher complexity applies to bounded classes of scalar-valued functions and i.i.d. datasets, which is not the case in our setting. To tackle this challenge, in the theorem above we employed the clipped component classes $\bar{\mathcal{F}}_{k,j}$ and the i.i.d. sub-datasets S_t. Employing a suitable contraction inequality from [31], we obtained the penalty (13), which depends on the empirical Rademacher complexities $\widehat{\mathcal{R}}_{S_t}(\bar{\mathcal{F}}_{k,j})$. As expected, the penalty (13) is more complicated than the one of classical SRM (see (4)).

In the following remark, we show that the SRM predictor of Theorem 2 guarantees a nearly optimal balance between approximation and estimation errors.

Remark 3 (SRM Guarantee for Nonlinear Dynamics)**.** *Suppose the true dynamics function $f_*(\cdot)$ is contained in $\mathcal{F} = \bigcup_{k=1}^{K} \mathcal{F}_k$. Let $k(f_*) := \min\{k : f_* \in \mathcal{F}_k\}$ be the index of the class that optimizes the approximation-estimation error trade-off. Moreover, let $f_S^{\mathrm{ERM}}(\cdot)$ be the ERM predictor in $\mathcal{F}_{k(f_*)}$. Then, from* (10) *and [28, Proposition 4.1] we obtain:*

$$\mathrm{er}_{\mathcal{D}}[\bar{f}_S^{\mathrm{ERM}}] \leq \mathrm{er}_{\mathcal{D}}[\bar{f}_*] + 2\,\epsilon_{k(f_*)}(S,\delta), \qquad (14)$$

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:27:49 UTC from IEEE Xplore. Restrictions apply.

whereas (12) *yields:*

$$\mathrm{er}_{\mathcal{D}}[\bar{f}_S^{\mathrm{SRM}}] \leq \mathrm{er}_{\mathcal{D}}[\bar{f}_*] + 2\,\epsilon_{k(f_*)}(S, (2\delta)/(K+1)). \quad (15)$$

From (14) *and* (15) *we conclude that the bound on the risk of the clipped SRM predictor $\bar{f}_S^{\mathrm{SRM}}(\cdot)$ is similar to that of the clipped ERM predictor $\bar{f}_S^{\mathrm{ERM}}(\cdot)$: the only difference is a factor of $2/(K+1)$ on the failure probability δ. Hence, modulo that factor, the SRM guarantee* (12) *is as favorable as the one we would have obtained, had an oracle informed us a priori of the optimal index $k(f_*)$.*

Beyond the above theoretical guarantee, SRM provides us with a model selection framework that applies to classes of arbitrary parametric model structures. The structural nature of SRM will become more clear in the next section, where explicit SRM schemes are provided for hierarchies of RKHS and NN classes.

IV. Explicit SRM for RKHS and NN Classes

The general SRM scheme we proposed in Section III for our setting of nonlinear dynamics consists of: i) picking a rich class $\mathcal{F} = \bigcup_{k=1}^{K} \mathcal{F}_k$ with $\mathcal{F}_k \subset \mathcal{F}_{k+1}$, $\forall k$, and ii) minimizing the sum of the training error (8) and the penalty (13), as dictated by (11). However, computing the penalty (13) is typically impossible in practice, as an exact expression of the empirical Rademacher complexities $\widehat{\mathcal{R}}_{S_t}(\bar{\mathcal{F}}_{k,j})$ is generally unavailable. To tackle this issue, we could replace $\widehat{\mathcal{R}}_{S_t}(\bar{\mathcal{F}}_{k,j})$'s with respective *computable* upper bounds. In this section, we employ such bounds and derive explicit SRM schemes for two large families of nonlinear functions: i) norm-constrained Reproducing Kernel Hilbert Spaces (RKHSs), and ii) norm-constrained Neural Network (NN) classes.

Before presenting our explicit SRM schemes, let us first introduce some notation. Consider the setting of Problem 1 and for each k, let $\bar{f}_k(\cdot)$ be the clipped version of $f_k := \arg\min_{f \in \mathcal{F}_k} \widehat{\mathrm{er}}_S[f]$. Herein, we refer to $\bar{f}_k(\cdot)$ as the *k-class predictor* of $\mathcal{F}$. For each k, let us also define the SRM error of $\bar{f}_k(\cdot)$ as $\mathrm{er}_{\mathrm{SRM}}[\bar{f}_k] = \widehat{\mathrm{er}}_S[\bar{f}_k] + r_k(S)$. Notice that the clipped SRM predictor is the k-class predictor with the minimum SRM error.

A. RKHS-based SRM for Nonlinear Dynamics

In this subsection, we introduce the norm-constrained RKHS [28] class employed in our setting as well as a corresponding SRM scheme.

Let $\kappa_\theta : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ be a continuous positive definite symmetric kernel [28, Definition 6.3] parametrized by a real vector θ. Examples of commonly used kernels [32] include the polynomial kernel $\kappa_{c,q}(x, x') := (x^\intercal x' + c)^q$, where $c \in \mathbb{R}_+$ and $q \geq 1$, and the Gaussian kernel $\kappa_\gamma(x, x') := \exp(-\gamma \|x - x'\|_2^2)$, where $\gamma > 0$. Let $\mathbb{H}$ denote the RKHS associated to κ_θ and $\|\cdot\|_{\mathbb{H}}$ its corresponding norm [28]. For any $\mathcal{B} > 0$, we define the *$\mathcal{B}$-constrained RKHS with kernel κ_θ* as $\mathcal{F}_{\mathrm{RKHS}}(\kappa_\theta, \mathcal{B}) = \left\{ f := \left[f_1, \ldots, f_n\right]^\intercal \mid f_j \in \mathbb{H}, \|f_j\|_{\mathbb{H}} \leq \mathcal{B}, \forall j \right\}$.

Consider a hierarchy of classes $\mathcal{F}_k = \mathcal{F}_{\mathrm{RKHS}}(\kappa_{\theta_k}, \mathcal{B}_k)$ of the above form. In that case, the SRM scheme of Theorem 2 cannot be applied directly, since an exact expression of $\widehat{\mathcal{R}}_{S_t}(\bar{\mathcal{F}}_{k,j})$'s is not available. However, we can upper-bound $\widehat{\mathcal{R}}_{S_t}(\bar{\mathcal{F}}_{k,j})$'s using standard computable bounds from [28]. Employing these bounds, we obtain the RKHS-based SRM scheme of the theorem below (see [27, Appendix C] for implementation details).

Theorem 3 (RKHS-based SRM for Nonlinear Dynamics)**.** *Fix a failure probability $\delta \in (0, 1)$. Let $\mathcal{F} = \bigcup_{k=1}^{K} \mathcal{F}_k$ with $\mathcal{F}_k = \mathcal{F}_{\mathrm{RKHS}}(\kappa_{\theta_k}, \mathcal{B}_k)$, $\forall k$, where the parameters κ_{θ_k}, $\mathcal{B}_k$ are such that $\mathcal{F}_k \subset \mathcal{F}_{k+1}$, $\forall k$. Let $\mathbb{H}_k$ be the RKHS associated to κ_{θ_k} and $\{M_1, \ldots, M_Q\}$ an increasing sequence of positive values with $M_Q = \max_k \mathcal{B}_k$. For each $f \in \mathcal{F}$, we define $q(f) = \min\left\{ q \,\middle|\, \max_j \|f_j\|_{\mathbb{H}_{k(f)}} \leq M_q \right\}$, where $k(f) = \min\{k \mid f \in \mathcal{F}_k\}$. Suppose the SRM predictor $f_S^{\mathrm{SRM}}(\cdot)$ is given by* (11) *with:*

$$r_k(S) = \frac{2\sqrt{2}LnM_{q(f_k)}}{TN} \sum_{t=0}^{T-1} \sqrt{\sum_{i=1}^{N} \kappa_{\theta_k}\left(x_t(\xi_i), x_t(\xi_i)\right)}. \quad (16)$$

Then, if $\epsilon_k(S, \delta) = \tilde{r}_k(S) + 6LB\sqrt{\log(4Q/\delta)/(2N)}$, where $\tilde{r}_k(S)$ is defined as $r_k(S)$ but with $M_{q(f_k)}$ replaced by $M_{\max\{q(f_k), q(f)\}}$, each of (10) *and* (12) *holds with probability at least $1-\delta$ over the draw of $S_0 \sim \mathcal{D}^N$.*

Note that the penalty (16) does not rely on the worst-case constraint constant $\mathcal{B}_k$ of the class $\mathcal{F}_k$, but on the discretized value $M_{q(f_k)}$ of the *learned* model $f_k(\cdot)$, which is selected from a pre-defined set $\{M_1, \ldots, M_Q\}$.

B. NN-based SRM for Nonlinear Dynamics

In this subsection, we present the class of neural networks [29] employed in our setting, along with a corresponding SRM scheme.

Let $g : \mathbb{R} \to \mathbb{R}$ be a 1-Lipschitz function that satisfies $g(az) = ag(z)$, for all $a \in \mathbb{R}_+$ and $z \in \mathbb{R}$ (e.g., consider the ReLU function $g(z) := \max\{0, z\}$, $\forall z$). Somewhat abusing notation, we will also allow $g(\cdot)$ to be applied element-wise (i.e., applied over each coordinate of its input). Let us also define $\tilde{g} = \begin{bmatrix} g & \mathbf{1} \end{bmatrix}^\intercal$, where $\mathbf{1}(\cdot)$ denotes the constant function equal to 1. Consider a neural network of the form:

$$f(x) = W_f^{(D_f)} \tilde{g}\left(W_f^{(D_f-1)} \tilde{g}\left(\cdots \tilde{g}\left(W_f^{(1)} \begin{bmatrix} x \\ 1 \end{bmatrix} \right) \right) \right), \quad (17)$$

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:27:49 UTC from IEEE Xplore. Restrictions apply.

where $x \in \mathbb{R}^n$ is the input, $D_f \geq 1$ is the depth, and $W_f^{(1)}, \ldots, W_f^{(D_f)}$ are the weight matrices of the neural network. Suppose we have $W_f^{(1)} \in \mathbb{R}^{(H_f+1)\times(n+1)}$, $W_f^{(D_f)} \in \mathbb{R}^{n\times(H_f+1)}$, and $W_f^{(d)} \in \mathbb{R}^{(H_f+1)\times(H_f+1)}$, for all $d = 2, \ldots, D_f - 1$, where $H_f \geq 1$ is the width of the neural network. Notice the increase of the input dimension n and the width H_f in the weight matrices $W_f^{(d)}$, so that biases are incorporated in them. For any $D \geq 1$, $H \geq 1$ and $\mathcal{B} > 0$, we define the *class of $\mathcal{B}$-constrained neural networks of depth D and width H* as:

$$\mathcal{F}_{\text{NN}}(D, H, \mathcal{B}) = \Big\{ x \mapsto f(x) \,\Big|\, f(\cdot) \text{ satisfies (17)}, \forall x, \\ D_f = D, H_f = H, \Big\| W_f^{(d)} \Big\|_{\text{F}} \leq \mathcal{B}, \forall d \Big\}. \tag{18}$$

In the case of NN classes $\mathcal{F}_k = \mathcal{F}_{\text{NN}}(D_k, H_k, \mathcal{B}_k)$ of the form (18), an exact expression of $\widehat{\mathcal{R}}_{S_t}(\bar{\mathcal{F}}_{k,j})$'s is not available. Therefore, the SRM scheme of Theorem 2 cannot be applied directly. Nevertheless, we can upper-bound $\widehat{\mathcal{R}}_{S_t}(\bar{\mathcal{F}}_{k,j})$'s using standard computable bounds from [33]. Employing theses bounds, we derive the NN-based SRM scheme presented in the following theorem.

Theorem 4 (NN-based SRM for Nonlinear Dynamics)**.** *Fix a failure probability $\delta \in (0,1)$. Let $\mathcal{F} = \bigcup_{k=1}^{K} \mathcal{F}_k$ with $\mathcal{F}_k = \mathcal{F}_{\text{NN}}(D_k, H_k, \mathcal{B}_k)$, $\forall k$, where the parameters D_k, H_k, $\mathcal{B}_k$ are such that $\mathcal{F}_k \subset \mathcal{F}_{k+1}$, $\forall k$. Moreover, let $\{M_1, \ldots, M_Q\}$ be an increasing sequence of positive values with $M_Q = \max_k \mathcal{B}_k$. For each $f \in \mathcal{F}$, we define $q(f) = \min \Big\{ q \,\Big|\, \max_d \Big\| W_f^{(d)} \Big\|_{\text{F}} \leq M_q \Big\}$. Suppose the SRM predictor $f_S^{\text{SRM}}(\cdot)$ is given by* (11) *with:*

$$r_k(S) = \frac{LnM_{q(f_k)}^{D_k} a(D_k)}{TN} \sum_{t=0}^{T-1} \sqrt{\sum_{i=1}^{N} (\|x_t(\xi_i))\|_2^2 + 1)}, \tag{19}$$

where $a(D_k) = 2\sqrt{2}\,(\sqrt{2\log(2)D_k} + 1)$. Then, if $\epsilon_k(S,\delta) = \tilde{r}_k(S) + 6LB\sqrt{\log(4Q/\delta)/(2N)}$, where $\tilde{r}_k(S)$ is defined as $r_k(S)$ but with $M_{q(f_k)}$ replaced by $M_{\max\{q(f_k),q(f)\}}$, each of (10) *and* (12) *holds with probability at least $1-\delta$ over the draw of $S_0 \sim \mathcal{D}^N$.*

Note that the penalty (19) does not rely on the worst-case constraint constant $\mathcal{B}_k$ of the class $\mathcal{F}_k$, but on the discretized value $M_{q(f_k)}$ of the *learned* model $f_k(\cdot)$, which is selected from a pre-defined set $\{M_1, \ldots, M_Q\}$.

Remark 4. *Unlike the general penalty* (13)*, the RKHS penalty* (16) *and the NN penalty* (19) *are both computable, thus providing us with the first explicit SRM schemes for learning of nonlinear dynamics. Note the implicit dependence of* (16) *on κ_{θ_k} and of* (19) *on H_k via the discretized value $M_{q(f_k)}$ of the respective learned model $f_k(\cdot)$. Similarly to our general SRM scheme, our RKHS- and NN-based SRM schemes guarantee a nearly optimal trade-off between approximation and estimation errors (see Remark 3). Due to discretization through the pre-specified set $\{M_1, \ldots, M_Q\}$, the corresponding error bounds $\epsilon_k(S,\delta)$ involve an extra factor of $1/Q$ on the failure probability δ. However, the failure probability does not affect the penalties, which in fact become tighter as Q increases. Hence, one could choose the discretized values $M_1, \ldots, M_{Q-1}$ by splitting $\big[0, M_Q\big]$ into sub-intervals of equal length, depending on the desired numerical accuracy.*

V. Experiments

In this section, we illustrate the efficacy of the proposed SRM schemes using two classical nonlinear systems. Specifically, in Subsection V-A, we estimate the double pendulum using our RKHS-based SRM scheme, whereas in Subsection V-B, we estimate the triple pendulum using our NN-based SRM scheme. Both pendula consist of identical compound pendula of length $l = 1$ and mass $m = 0.5$. Furthermore, in each case, we obtain the corresponding discretization set $\{M_1, \ldots, M_Q\}$ by splitting $\big[0, M_Q\big]$ into sub-intervals of length 0.01.

A. Double Pendulum

Assume the double pendulum's motion is modeled via Hamilton's equations [34], with each component of the initial state independently uniformly distributed in $\big[-0.5, 0.5\big]$. The system is 4-dimensional with the first two states representing the angular position and the last two states the momentum [34]. The bound B on the system state norm was roughly computed as $B = 2.2$ (see Remark 1). Suppose we are given a dataset S of $N = 140$ trajectories of length $T = 5$. We generated these trajectories by sampling the original (continuous-time) trajectories with period $T_s = 0.05s$. Employing the dataset S and the loss function $\|\cdot\|_2$, our goal is to estimate the discretized model $f_*(\cdot)$ using our RKHS-based SRM scheme. In particular, consider Gaussian kernels with parameters $\gamma_k = 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 5, 10$ and define the class $\mathcal{F} = \bigcup_{k=1}^{8} \mathcal{F}_k$ with $\mathcal{F}_k = \mathcal{F}_{\text{RKHS}}(\kappa_{\gamma_k}, 20)$, $\forall k$. Notice that the complexity of $\mathcal{F}_k$ increases with k, despite the fact that the classes $\mathcal{F}_k$ are not nested (see Remark 2). This is because smaller values of γ produce Gaussian kernels with smoother decision boundaries.

Figure 2 shows the training, SRM and true errors of the k-class predictors $\bar{f}_k(\cdot)$ learned from the RKHS classes $\mathcal{F}_k$. We estimated the true error using 10^4 test trajectories. We observe that the true error decreases for

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:27:49 UTC from IEEE Xplore. Restrictions apply.

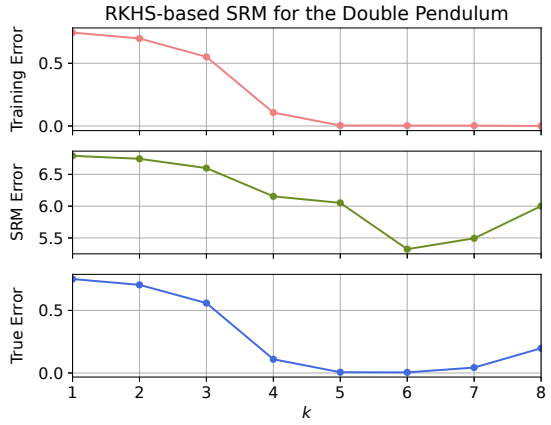


Fig. 2. The training error (top), the SRM error (middle), and the true error (bottom) of the k-class predictors $\bar{f}_k(\cdot)$ corresponding to the RKHS classes $\mathcal{F}_{\text{RKHS}}(\kappa_{\gamma_k}, 20)$. The training error fails to track the increase of the true error for $k > 6$, whereas the SRM error achieves tracking the relative behavior of the true error across all k's.

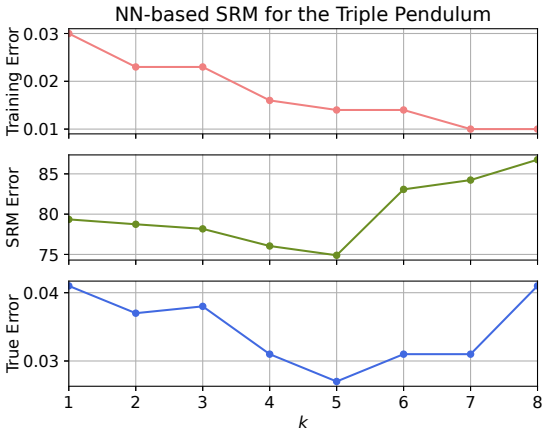


Fig. 3. The training error (top), the SRM error (middle), and the true error (bottom) of the k-class predictors $\bar{f}_k(\cdot)$ corresponding to the NN classes $\mathcal{F}_{\text{NN}}(2, H_k, 12)$. The training error fails to track the increase of the true error for $k > 5$. In contrast, the SRM error achieves tracking the relative behavior of the true error across all k's (except for $k = 3$ where a negligible increase of the true error is observed).

$k < 6$ and increases for $k > 6$. This was expected since smaller γ's lead to simpler classes that may not capture the complexity of $f_*(\cdot)$, whereas larger γ's lead to more complex classes where learning is more difficult. We can see that the training error keeps decreasing over k, thus failing to follow the true error for $k > 6$. In contrast, the SRM error tracks the relative behavior of the true error for all k. Particularly, note that the minimum SRM and true error are both attained at $\gamma_6 = 1$, which yields a risk of $\text{er}_{\mathcal{D}}[\bar{f}_S^{\text{SRM}}] \approx 0.006$.

B. Triple Pendulum

Assume the triple pendulum's motion is modeled via Kane's equations [35]. The system is 6-dimensional with the first three states describing the angular position and the last three states the angular velocity. The initial state is drawn from a zero-mean Gaussian truncated at $B_\xi := \{x \in \mathbb{R}^6 \,|\, |x_i| \leq \pi/4, i = 1, 2, 3, |x_i| \leq 0.5, i = 4, 5, 6\}$. The bound on the system state norm was roughly computed as $B = 4$ (see Remark 1). Suppose we are given a dataset S of $N = 150$ trajectories of length $T = 5$. We generated these trajectories by sampling the original (continuous-time) trajectories with period $T_s = 0.05$. Employing the dataset S and the loss function $\|\cdot\|_2$, we wish to estimate the discretized model $f_*(\cdot)$ using our NN-based SRM scheme. Specifically, consider neural networks with ReLU activation, constant depth $D = 2$, and increasing width $H_k = 25, 35, \ldots, 95$, and define the class $\mathcal{F} = \bigcup_{k=1}^{8} \mathcal{F}_k$ with $\mathcal{F}_k = \mathcal{F}_{\text{NN}}(2, H_k, 12)$, $\forall k$.

Figure 3 shows the training, SRM and true errors of the k-class predictors $\bar{f}_k(\cdot)$ learned from the NN classes $\mathcal{F}_k$. We implemented the norm-constrained NN classes as in [36] and estimated the true error using a dataset of 10^4 test trajectories. Notice that the true error decreases for $k < 5$ and increases for $k > 5$ (we will ignore the negligible increase at $k = 3$). This is because neural networks with smaller width may not capture the complexity of $f_*(\cdot)$, whereas neural networks with larger width are more difficult to learn. We observe that the training error decreases for the entire range of NN widths, thus failing to follow the true error for $k > 5$. Unlike the training error, the SRM error achieves tracking the relative behavior of the true error for all k. Particularly, note that the minimum SRM and true error are both attained at $H_5 = 65$, which yields a risk of $\text{er}_{\mathcal{D}}[\bar{f}_S^{\text{SRM}}] \approx 0.027$.

Discussion: In both experiments we observe that the values of the SRM error are of larger order than the values of the training and true error. Despite being too loose to be used as absolute estimates of the true error, our SRM bounds remarkably succeed in tracking its relative behavior. Figures 2 and 3 show that SRM attains in fact an optimal approximation-estimation error trade-off, thus validating our learning guarantee (see Remark 3). This can be explained by the dependence of the penalties (16), (19) on the *learned* k-class predictors $\bar{f}_k(\cdot)$ via the discretized term $M_{q(f_k)}$ (see Remark 4).

VI. Future Work

Going forward, our paper opens up several research directions. First, in order to improve the statistical effi-

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:27:49 UTC from IEEE Xplore. Restrictions apply.

ciency of our SRM scheme, we could exploit bounds for dependent data to analyze each trajectory. Moreover, an extension of our SRM method to nonlinear dynamical systems with additive noise and external inputs could be considered. Finally, it would be interesting to provide explicit SRM schemes for other model classes, such as Recurrent Neural Networks (RNNs).

REFERENCES

[1] J. D. Anderson and J. Wendt, *Computational fluid dynamics*. Springer, 1995, vol. 206.

[2] A. Rajkomar, J. Dean, and I. Kohane, "Machine learning in medicine," *New England Journal of Medicine*, vol. 380, no. 14, pp. 1347–1358, 2019.

[3] R. H. Day, *Complex Economic Dynamics - Vol. 1: An Introduction to Dynamical Systems and Market Mechanisms*, ser. MIT Press Books. The MIT Press, 1994, vol. 1.

[4] Z. Ghahramani and S. Roweis, "Learning nonlinear dynamical systems using an em algorithm," *Advances in neural information processing systems*, vol. 11, 1998.

[5] J. Langford, R. Salakhutdinov, and T. Zhang, "Learning nonlinear dynamic models," in *Proceedings of the 26th Annual International Conference on Machine Learning*, 2009, pp. 593–600.

[6] S. M. Khansari-Zadeh and A. Billard, "Learning stable nonlinear dynamical systems with gaussian mixture models," *IEEE Transactions on Robotics*, vol. 27, no. 5, pp. 943–957, 2011.

[7] G. Pillonetto, F. Dinuzzo, T. Chen, G. De Nicolao, and L. Ljung, "Kernel methods in system identification, machine learning and function estimation: A survey," *Automatica*, vol. 50, no. 3, pp. 657–682, 2014.

[8] A. Svensson and T. B. Schön, "A flexible state–space model for learning nonlinear dynamical systems," *Automatica*, vol. 80, pp. 189–199, 2017.

[9] K. Neumann, A. Lemme, and J. J. Steil, "Neural learning of stable dynamical systems based on data-driven lyapunov candidates," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1216–1222.

[10] M. Raissi, P. Perdikaris, and G. E. Karniadakis, "Multistep neural networks for data-driven discovery of nonlinear dynamical systems," *arXiv preprint arXiv:1801.01236*, 2018.

[11] Q. Teng and L. Zhang, "Data driven nonlinear dynamical systems identification using multi-step cldnn," *AIP Advances*, vol. 9, no. 8, p. 085311, 2019.

[12] T. Qin, Z. Chen, J. D. Jakeman, and D. Xiu, "Deep learning of parameterized equations with applications to uncertainty quantification," *International Journal for Uncertainty Quantification*, vol. 11, no. 2, 2021.

[13] D. Foster, T. Sarkar, and A. Rakhlin, "Learning nonlinear dynamical systems from a single trajectory," in *Learning for Dynamics and Control*, 2020, pp. 851–861.

[14] S. Singh, S. M. Richards, V. Sindhwani, J.-J. E. Slotine, and M. Pavone, "Learning stabilizable nonlinear dynamics with contraction-based regularization," *The International Journal of Robotics Research*, vol. 40, no. 10-11, pp. 1123–1150, 2021.

[15] S. L. Brunton, J. L. Proctor, and J. N. Kutz, "Discovering governing equations from data by sparse identification of nonlinear dynamical systems," *Proceedings of the national academy of sciences*, vol. 113, no. 15, pp. 3932–3937, 2016.

[16] N. M. Mangan, J. N. Kutz, S. L. Brunton, and J. L. Proctor, "Model selection for dynamical systems via sparse regression and information criteria," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 473, no. 2204, p. 20170009, 2017.

[17] K. Champion, B. Lusch, J. N. Kutz, and S. L. Brunton, "Data-driven discovery of coordinates and governing equations," *Proceedings of the National Academy of Sciences*, vol. 116, no. 45, pp. 22 445–22 451, 2019.

[18] B. Lusch, J. N. Kutz, and S. L. Brunton, "Deep learning for universal linear embeddings of nonlinear dynamics," *Nature communications*, vol. 9, no. 1, p. 4950, 2018.

[19] E. Kaiser, J. N. Kutz, and S. L. Brunton, "Data-driven discovery of koopman eigenfunctions for control," *Machine Learning: Science and Technology*, vol. 2, no. 3, p. 035023, 2021.

[20] G. Lugosi and K. Zeger, "Concept learning using complexity regularization," *IEEE Transactions on Information Theory*, vol. 42, no. 1, pp. 48–54, 1996.

[21] V. Koltchinskii, "Rademacher penalties and structural risk minimization," *IEEE Transactions on Information Theory*, vol. 47, no. 5, pp. 1902–1914, 2001.

[22] J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, and M. Anthony, "Structural risk minimization over data-dependent hierarchies," *IEEE transactions on Information Theory*, vol. 44, no. 5, pp. 1926–1940, 1998.

[23] C. Gyurik, V. Dunjko, *et al.*, "Structural risk minimization for quantum linear classifiers," *Quantum*, vol. 7, p. 893, 2023.

[24] J. Liu, M. Bai, N. Jiang, and D. Yu, "Structural risk minimization of rough set-based classifier," *Soft Computing*, vol. 24, pp. 2049–2066, 2020.

[25] L. Massucci, F. Lauer, and M. Gilson, "Structural risk minimization for switched system identification," in *2020 59th IEEE Conference on Decision and Control (CDC)*, 2020, pp. 1002–1007.

[26] R. Meir, "Structural risk minimization for nonparametric time series prediction," *Advances in Neural Information Processing Systems*, vol. 10, 1997.

[27] C. Stamouli, E. Chatzipantazis, and G. J. Pappas, "Structural risk minimization for learning nonlinear dynamics," *arXiv preprint arXiv:2309.16527*, 2023.

[28] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*. MIT Press, 2nd edition, 2018.

[29] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[30] I. M. Ziemann, H. Sandberg, and N. Matni, "Single trajectory nonparametric learning of nonlinear dynamics," in *conference on Learning Theory*, 2022, pp. 3333–3364.

[31] C. Cortes, V. Kuznetsov, M. Mohri, and S. Yang, "Structured prediction theory based on factor graph complexity," *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[32] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[33] N. Golowich, A. Rakhlin, and O. Shamir, "Size-independent sample complexity of neural networks," in *Proceedings of the 31st Conference On Learning Theory*, vol. 75, 2018, pp. 297–299.

[34] B. Stickler and E. Schachinger, *Basics Concepts in Computational Physics*. Springer, 2016.

[35] G. Gede, D. L. Peterson, A. S. Nanjangud, J. K. Moore, and M. Hubbard, "Constrained multibody dynamics with python: From symbolic equation generation to publication," in *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, vol. 55973, 2013, p. V07BT10A051.

[36] H. Gouk, E. Frank, B. Pfahringer, and M. J. Cree, "Regularisation of neural networks by enforcing lipschitz continuity," *Machine Learning*, vol. 110, pp. 393–416, 2021.

Authorized licensed use limited to: University of Pennsylvania. Downloaded on October 29,2026 at 14:27:49 UTC from IEEE Xplore. Restrictions apply.