Generalizing Dynamic Mode Decomposition: Balancing Accuracy and Expressiveness in Koopman Approximations

Masih Haseli [a] Jorge Cortés [a]

[a] *Department of Mechanical and Aerospace Engineering, University of California, San Diego, {mhaseli,cortes}@ucsd.edu*

Abstract

This paper tackles the data-driven approximation of unknown dynamical systems using Koopman-operator methods. Given a dictionary of functions, these methods approximate the projection of the action of the operator on the finite-dimensional subspace spanned by the dictionary. We propose the Tunable Symmetric Subspace Decomposition algorithm to refine the dictionary, balancing its expressiveness and accuracy. Expressiveness corresponds to the ability of the dictionary to describe the evolution of as many observables as possible and accuracy corresponds to the ability to correctly predict their evolution. Based on the observation that Koopman-invariant subspaces give rise to exact predictions, we reason that prediction accuracy is a function of the degree of invariance of the subspace generated by the dictionary and provide a data-driven measure to measure invariance proximity. The proposed algorithm iteratively prunes the initial functional space to identify a refined dictionary of functions that satisfies the desired level of accuracy while retaining as much of the original expressiveness as possible. We provide a full characterization of the algorithm properties and show that it generalizes both Extended Dynamic Mode Decomposition and Symmetric Subspace Decomposition. Simulations on planar systems show the effectiveness of the proposed methods in producing Koopman approximations of tunable accuracy that capture relevant information about the dynamical system.

1 Introduction

Progress in data acquisition and labeling, along with widespread access to high-performance computing capabilities for storing, processing, and data analysis, has resulted in a surge of activity in learning and modeling of dynamical phenomena across multiple domains. In this context, the importance of identification techniques that yield tractable representations of nonlinear dynamics rooted at a solid theoretical framework cannot be overemphasized. One such technique is Koopman operator theory which, instead of prescribing the evolution of system trajectories as state-space models do, describes the evolution of functions defined over the state space (a.k.a. observables). The infinite-dimensional nature of the operator has prevented its widespread use due to the lack of computational methods to represent it. Extended dynamic mode decomposition (EDMD) addresses this by employing data from the dynamics to approximate the projection of the action of the Koopman operator on a finite-dimensional subspace spanned by a predefined dictionary of functions.

Despite EDMD's success, it is still not well understood how to choose dictionaries that both capture relevant information about the dynamics and are able to accurately predict its evolution. Prediction accuracy is related to the degree of invariance, with respect to the operator, of the subspace generated by the dictionary and, in fact, can be improved by selectively pruning its functions. Such process, however, impacts expressiveness, understood as the ability of the dictionary to describe the evolution of as many observables as possible. This paper is motivated by the need to address the accuracy-expressiveness trade-off in dictionary selection.

Literature Review: Given a dynamical system, its associated Koopman operator [19, 20] is a linear operator characterizing the effect of the dynamics on functions in a (generally infinite-dimensional) linear functional space. The values of its eigenfunctions also evolve linearly in time on the trajectories of the system. These properties enable the use of the spectral properties of the operator to process data from dynamical systems using efficient linear algebraic methods fully compatible with the arithmetic operations of digital computers [5, 31]. This has led to many applications in fluid dynamics [36], model reduction [31], and control, including controller synthesis [6, 9, 10], model predictive control [21, 38], control of PDEs [35], and robotic applications [26, 45].

The infinite-dimensional nature of the Koopman operator is an impediment to its direct implementation on digital computers. A popular way to find finite-dimensional representations of the operator is through Dynamic Mode Decomposition (DMD) and its variants, initially developed to extract dynamical information from data about fluid flows [37, 41]. [42] developed the Extended Dynamic Mode Decomposition (EDMD) algorithm, a variant of DMD capable of approximating the projection of the action of the Koopman operator from data on a finite-dimensional space spanned by a chosen dictionary of functions. [22] formally established the convergence of the EDMD approximation to the projection of the action of the operator on the span of the dictionary. Recently, [24] analyzed the accuracy of long-term prediction by DMD and its variants. [27] used a Taylor expansion method to enrich the dictionary for EDMD to achieve lower errors in long-term predictions. [44] used finite element methods to learn Koopman approximations with accuracy bounds quan-

* This work was supported by ONR Award N00014-18-1-2828 and NSF Award IIS-2007141.

A preliminary version of this paper appeared at the American Control Conference as [12].

tifying their quality. This work also provided a variant of EDMD to learn the Koopman generator combined with finite element methods. [32] provided several probabilistic bounds for approximation of the Koopman operator based on the availability of only finitely many data points both for deterministic and stochastic systems. It is worth mentioning that (E)DMD captures valuable information about stochastic dynamical systems [18, 42]; however, it is sensitive to the existence of measurement noise in the available data. [8] and [11] provide methods to deal with measurement noise in data for DMD and EDMD, respectively.

In general, given an arbitrary dictionary, there is no guarantee that EDMD provides an accurate approximation for all the observables in the span of the dictionary. This has resulted in the search for dictionaries that span invariant subspaces [3] under the Koopman operator, on which the EDMD approximation is exact. The work [15] introduces a class of logistic functions to approximate Koopman-invariant subspaces for systems whose dynamics are known. On the other hand, given unknown systems and using data sampled from their trajectories, Koopman-invariant subspaces are approximated using neural networks in [25, 33, 39, 43] and sparsity-promoting methods in [34]. The works by [16, 23] directly approximate Koopman eigenfunctions, which naturally span Koopman-invariant subspaces. These data-driven methods do not provide theoretical guarantees for the resulting approximations. Given the importance of such guarantees, our previous work [13, 14] has provided necessary and almost surely sufficient conditions for the identification of the maximal Koopman-invariant subspace and all Koopman eigenfunctions in an arbitrary finite-dimensional functional space. We have also provided approximations to identify subspaces that are close to being invariant for cases when the maximal Koopman-invariant subspace does not capture enough information about the dynamics.

It is important to note that the existence of finite-dimensional Koopman-invariant subspaces containing the states of the system is not guaranteed [3]. However, invariant subspaces still contain Koopman eigenfunctions, and can capture relevant information about the vector field, physical constraints, conservation laws, stability, and even the construction of Lyapunov functions, see e.g., [30]. Nonetheless, in some applications, one can tolerate a certain level of inaccuracy in order to capture a more diverse functional space that is not necessarily Koopman invariant but captures important variables such as the states of the system.

Statement of Contributions: We consider the problem of data-driven identification of finite-dimensional spaces that are close, with tunable accuracy, to being invariant under the action of the Koopman operator. Our main result, illustrated in Figure 1, consists of the synthesis of a computational procedure, termed Tunable Symmetric Subspace Decomposition (T-SSD), that given an *arbitrary* finite-dimensional functional space, balances the trade-off between the expressiveness [1] of its subspaces and the accuracy of the Koopman approximations on them.

[1] We note that notions of expresiveness different that the one adopted here are also possible. For instance, and although out of the scope of the paper, expresiveness could also be understood as the ability of the dictionary to describe specific finitely many predefined observables of interest, such as state variables, in a *given* set of coordinates.

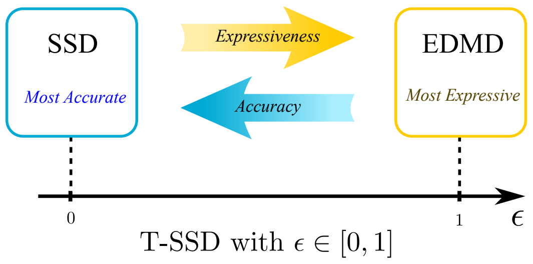

Figure 1. T-SSD generalizes SSD and EDMD. Given an *arbitrary* finite-dimensional functional space, by changing the design parameter $\epsilon \in [0,1]$ in T-SSD, one can strike a balance between the invariance proximity of the identified subspace (i.e., the accuracy of the resulting model based on the available data) and its expressiveness.

The roadmap of supporting contributions leading to the design and full characterization of T-SSD is as follows. Our first contribution builds on the observation that the proximity of a functional space to being invariant is a measure of its (and consequently its members') prediction accuracy under finite-dimensional Koopman approximations, as an exact invariant subspace leads to exact predictions of the evolution of observables. We introduce the novel notion of ϵ-apart spaces to measure invariance proximity using data snapshots sampled from the trajectories of the unknown dynamics. Using this notion, and given an arbitrary finite-dimensional functional space spanned by a dictionary of functions, we formulate our objective as that of finding a parametric family of subspaces whose value of the parameter determines the desired level of invariance proximity. This parametric family can be viewed as balancing invariance proximity (i.e., prediction accuracy) and the dimension of the subspace (i.e., expressiveness).

Given a desired accuracy parameter, our second contribution is the design of T-SSD as an algorithmic procedure that finds a functional space satisfying the desired accuracy by iteratively removing the functions in the span of the original dictionary that violate the desired accuracy. We show that T-SSD terminates in a finite number of iterations and characterize its computational complexity. Moreover, we show that its identified subspaces contain the maximal Koopman-invariant subspace and all Koopman eigenfunctions in the span of the original dictionary. We also show that the accuracy parameter bounds the relative root mean square prediction error for all (uncountably many) functions in the identified subspace. This advantage of the T-SSD algorithm in deriving accuracy bounds on the prediction of individual functions independently of linear changes of coordinates stems from focusing on the subspaces instead of their basis. Our next contribution establishes that both Extended Dynamic Mode Decomposition and Symmetric Subspace Decomposition algorithms are particular cases of T-SSD, cf. Figure 1.

Our final contribution is a computationally efficient version of T-SSD with drastically lower computational complexity when the number of data snapshots is significantly larger than the dimension of the original dictionary. We illustrate in simulation the effectiveness of T-SSD in identifying informative Koopman approximations of tunable accuracy.

2 Preliminaries

Here, we introduce the notation used in the paper and provide a brief overview of Koopman operator theory, extended dynamic mode decomposition (EDMD), and symmetric subspace decomposition (SSD).

2.1 Notation

We let $\mathbb{R}$, $\mathbb{C}$, $\mathbb{N}_0$, and $\mathbb{N}$ represent the sets of real, complex, nonnegative integer, and natural numbers respectively. Given a matrix $A \in \mathbb{C}^{m\times n}$, we denote its transpose, conjugate transpose, pseudo-inverse, range space, and Frobenius norm by A^T, A^H, $A^\dagger$, $\mathcal{R}(A)$, and $\|A\|_F$ respectively. In addition, cols(A), rows(A), $\sharp$cols(A), and $\sharp$rows(A) represent its set of columns, set of rows, number of columns, and number of rows. Also if A is a nonsingular square matrix, A^{-1} denotes its inverse. Given $A \in \mathbb{C}^{m\times n}$ and $B \in \mathbb{C}^{m\times p}$, we use $[A, B]$ to represent the matrix formed by their side by side concatenation. We denote by $\mathbf{0}_{m\times n}$ and I_n, the $m \times n$ zero matrix and the identity matrix of size n respectively (we omit the indices when the context is clear). Given a vector $v \in \mathbb{C}^n$, $\|v\|_2 := \sqrt{v^H v}$ denotes its 2-norm.

We denote by span$\{v_1, \ldots, v_k\}$ the vector space over $\mathbb{C}$ spanned by $v_1, \ldots, v_k \in \mathbb{C}^n$. Given functions $f_1, \ldots, f_k$, span$\{f_1, \ldots, f_k\}$ is the functional space formed by all functions in the form of $c_1 f_1 + \cdots + c_k f_k$ with $\{c_i\}_{i=1}^k \subset \mathbb{C}$. For a vector space $\mathcal{V} \subseteq \mathbb{R}^m$, $\mathcal{P}_\mathcal{V}$ denotes the orthogonal projection operator on $\mathcal{V}$. For convenience, we denote the orthogonal projection operator on the range space of a matrix A by $\mathcal{P}_A$, which takes the form $\mathcal{P}_A\, w = AA^\dagger w$, for $w \in \mathbb{R}^m$. For vectors $v, w \in \mathbb{R}^m$, $v \perp w$ indicates that v and w are orthogonal. Given vector spaces $\mathcal{V}_1, \mathcal{V}_2 \subseteq \mathbb{R}^m$, $\mathcal{V}_1 \perp \mathcal{V}_2$ denotes that the vector spaces are orthogonal, i.e., all vectors in $\mathcal{V}_1$ are orthogonal to all vectors in $\mathcal{V}_2$. We define the sum of vector spaces $\mathcal{V}_1, \mathcal{V}_2$ by $\mathcal{V}_1 + \mathcal{V}_2 := \{v_1 + v_2 | v_1 \in \mathcal{V}_1 \wedge v_2 \in \mathcal{V}_2\}$. Given sets S_1, S_2, we denote their union and intersection by $S_1 \cup S_2$ and $S_1 \cap S_2$ respectively. Given functions $f : S_2 \to S_3$, $g : S_1 \to S_2$, $f \circ g : S_1 \to S_3$ denotes their composition.

2.2 Koopman Operator

Our exposition follows [5]: given the discrete-time dynamics

$$x^+ = T(x). \tag{1}$$

defined on the state space $\mathcal{M} \subseteq \mathbb{R}^n$ and a linear space of functions $\mathcal{F}$ defined over the field $\mathbb{C}$ and closed under composition with T, the Koopman operator $\mathcal{K} : \mathcal{F} \to \mathcal{F}$ associated with (1) is defined as $\mathcal{K}f = f \circ T$. The functions in $\mathcal{F}$ are called *observables*. As a consequence of the linearity of $\mathcal{F}$, the Koopman operator is *spatially linear*, i.e.,

$$\mathcal{K}(c_1 f_1 + c_2 f_2) = c_1\mathcal{K}(f_1) + c_2\mathcal{K}(f_2), \tag{2}$$

for any $f_1, f_2 \in \mathcal{F}$ and $c_1, c_2 \in \mathbb{C}$. The linearity of the Koopman operator paves the way for defining its eigendecomposition. Formally, a function $\phi \in \mathcal{F}$ is an *eigenfunction* of the Koopman operator with *eigenvalue* $\lambda \in \mathbb{C}$ if

$$\mathcal{K}\phi = \lambda\phi. \tag{3}$$

The eigenfunctions of the Koopman operator evolve linearly in time on the trajectories of (1),

$$\phi(x^+) = \phi \circ T(x) = \mathcal{K}\phi(x) = \lambda\phi(x). \tag{4}$$

The linear temporal evolution of eigenfunctions combined with (2) make the Koopman operator a powerful tool for *linear prediction* of the functions' values on the trajectories of the (generally nonlinear) dynamical system (1). Formally, given a function $f = \sum_{i=1}^{N_k} c_i\phi_i$, where $\{c_i\}_{i=1}^{N_k} \subset \mathbb{C}$ and $\{\phi_i\}_{i=1}^{N_k}$ are eigenfunctions of $\mathcal{K}$ with eigenvalues $\{\lambda_i\}_{i=1}^{N_k}$, one can predict the values of f on the trajectory starting from initial condition $x_0 \in \mathcal{M}$ as

$$f(x(k)) = \sum_{i=1}^{N_k} c_i\lambda_i^k\, \phi_i(x(0)), \quad \forall k \in \mathbb{N}. \tag{5}$$

Finally, a subspace $\mathcal{L} \subseteq \mathcal{F}$ is *Koopman-invariant* if for every $f \in \mathcal{L}$, we have $\mathcal{K}f \in \mathcal{L}$. Trivially, any subspace spanned by Koopman eigenfunctions is Koopman invariant.

2.3 Extended Dynamic Mode Decomposition

In general, the Koopman operator is infinite dimensional. Moreover, in many practical data-driven applications, the dynamical system is unknown and only data from some trajectories is available. These issues motivate the use of data-driven methods to approximate the effect of the Koopman operator on finite-dimensional subspaces, as we discuss next. Here, we recall the Extended Dynamic Mode Decomposition (EDMD) method following [42]. EDMD uses data from the trajectories of the system to approximate the action of the Koopman operator on a predefined functional space. The first ingredient of EDMD is the data matrices $X, Y \in \mathbb{R}^{N\times n}$ containing N data snapshots, where corresponding rows of X, Y characterize two consecutive points on a trajectory of the system. Formally,

$$y_i = T(x_i), \forall i \in \{1, \ldots, N\}, \tag{6}$$

where x_i^T and y_i^T are the ith rows of X and Y respectively. The second ingredient of EDMD is a dictionary $D : \mathcal{M} \to \mathbb{R}^{1\times N_d}$ of N_d functions, denoted as

$$D(x) = [d_1(x), \ldots, d_{N_d}(x)], \tag{7}$$

where $d_i : \mathcal{M} \to \mathbb{R}$ for all $i \in \{1, \ldots, N_d\}$. The dictionary spans a finite-dimensional space of functions over $\mathbb{C}$, and its elements can be complex-valued functions in general. However, since the system is defined over the state space $\mathcal{M} \subset \mathbb{R}^n$, complex Koopman eigenfunctions form complex conjugate pairs which can be fully represented by a pair of real-valued functions. For this reason, and without loss of generality, we use real-valued functions as dictionary elements.

EDMD formulates a least-squares optimization to find the best fit for the linear evolution of the dictionary functions on the data,

$$\underset{K}{\text{minimize}} \|D(Y) - D(X)K\|_F. \tag{8}$$

whose closed-form solution is

$$K_{\text{EDMD}} = \text{EDMD}(D, X, Y) := D(X)^\dagger D(Y). \tag{9}$$

K_{EDMD} approximates the projection of the action of the Koopman operator on span(D) as follows. Under the identification of $\mathbb{C}^{N_d}$ with span(D) defined by $v \leftrightarrow D(\cdot)v$, this approximation takes the form

$$v \mapsto K_{\text{EDMD}} v. \tag{10}$$

As a result, for the function $f(\cdot) = D(\cdot)v_f \in \text{span}(D)$, one can define the EDMD's approximation of $\mathcal{K}f$ as

$$\mathfrak{P}_{\mathcal{K}f} = D(\cdot)K_{\text{EDMD}}v_f. \tag{11}$$

It is important to note that $\mathfrak{P}_{\mathcal{K}f}$ can be viewed as the L_2-orthogonal projection of $\mathcal{K}f$ on span(D) given an empirical measure defined based on the rows of X [17, 22]. Moreover, the eigendecomposition of K_{EDMD} provides insight into the eigendecomposition of the Koopman operator. Formally, given an eigenpair (λ, v) of K_{EDMD}, one approximates an eigenfunction of the Koopman operator with eigenvalue λ as

$$\phi(\cdot) := D(\cdot)v. \tag{12}$$

Note that the EDMD predictor in (11) applied to the eigenfunction ϕ in (12) leads to $\mathfrak{P}_{\mathcal{K}\phi} = \lambda\phi$, which is in agreement with the linear evolution (4) of Koopman eigenfunctions. Note that EDMD provides N_d Koopman (generalized) eigenfunction approximations even if the Koopman operator does not have N_d eigenfunctions in the span of the dictionary. We refer the reader to [22] for *asymptotic* convergence results as $N_d \to \infty$, concerning K_{EDMD} capturing the spectrum of the Koopman operator as well as the finite-horizon prediction of EDMD for observables in span(D).

If the dictionary D spans a Koopman-invariant subspace, then EDMD completely captures the behavior of the Koopman operator on span(D). As a result, $\|D(Y) - D(X)K_{\text{EDMD}}\|_F = 0$, and (11) provides exact prediction, independently of the data used for EDMD's training. A final note regarding EDMD is that it is not designed to work with data corrupted with measurement noise. Hence, data pre-processing might be needed to take full advantage of the methods proposed in the paper, which rely on EDMD.

2.4 Symmetric Subspace Decomposition

In general, since the true system dynamics is unknown, choosing a dictionary that spans a Koopman-invariant subspace is challenging. This justifies the importance of developing data-driven methods that aid in this task. Here, we recall the Symmetric Subspace Decomposition (SSD) algorithm following [14]. Starting from a dictionary D and data snapshots X, Y, the SSD algorithm, cf. Algorithm 1, finds a basis for the maximal Koopman-invariant subspace in span(D). To achieve this goal, SSD relies on the following.

Assumption 2.1 *(Full Rank Dictionary Matrices): The matrices $D(X)$ and $D(Y)$ have full column rank.* □

Assumption 2.1 rules out the case of redundant dictionary elements by making sure its functions are linearly independent. Moreover, it requires the data snapshots to be sampled in a way that reflects this fact.

The SSD algorithm provides an iterative approach to prune the dictionary D at each iteration by removing the functions that do not correspond to linear evolutions. The SSD algorithm terminates after finite iterations and its output C_{SSD} satisfies the following properties.

Theorem 2.2 *(SSD Output [14, Theorem 5.1]): Suppose that Assumption 2.1 holds. Then,*

(a) *C_{SSD} is either 0 or has full column rank;*
(b) *C_{SSD} satisfies $\mathcal{R}(D(X)C_{\text{SSD}}) = \mathcal{R}(D(Y)C_{\text{SSD}})$;*
(c) *the subspace $\mathcal{R}(D(X)C_{\text{SSD}})$ is maximal, in the sense that, for any matrix E with $\mathcal{R}(D(X)E) = \mathcal{R}(D(Y)E)$, we have $\mathcal{R}(D(X)E) \subseteq \mathcal{R}(D(X)C_{\text{SSD}})$ and $\mathcal{R}(E) \subseteq \mathcal{R}(C_{\text{SSD}})$.* □

Algorithm 1 Symmetric Subspace Decomposition

```
    Inputs: $D(X), D(Y) \in \mathbb{R}^{N \times N_d}$    Output: $C_{\text{SSD}}$
 1: Procedure $C_{\text{SSD}} \leftarrow \text{SSD}(D(X), D(Y))$
 2: Initialization
 3: $i \leftarrow 1$, $A_1 \leftarrow D(X)$, $B_1 \leftarrow D(Y)$, $C_{\text{SSD}} \leftarrow I_{N_d}$
 4: while 1 do
 5:     $\begin{bmatrix} Z_i^A \\ Z_i^B \end{bmatrix} \leftarrow \text{null}([A_i, B_i])$    ▷ Basis for the null space
 6:     if $\text{null}([A_i, B_i]) = \emptyset$ then
 7:         return 0    ▷ The basis does not exist
 8:         break
 9:     end if
10:     if $\sharp\text{rows}(Z_i^A) \le \sharp\text{cols}(Z_i^A)$ then
11:         return $C_{\text{SSD}}$    ▷ The procedure is complete
12:         break
13:     end if
14:     $C_{\text{SSD}} \leftarrow C_{\text{SSD}} Z_i^A$    ▷ Reduce the subspace
15:     $A_{i+1} \leftarrow A_i Z_i^A$, $B_{i+1} \leftarrow B_i Z_i^A$, $i \leftarrow i + 1$
16: end while
```

The dictionary identified by SSD is

$$D_{\text{SSD}}(\cdot) := D(\cdot)C_{\text{SSD}}. \tag{13}$$

Based on Theorem 2.2, D_{SSD} spans the largest subspace of span(D) on which $\mathcal{R}(D_{\text{SSD}}(X)) = \mathcal{R}(D_{\text{SSD}}(Y))$. One can apply the EDMD algorithm (equations (8)-(9)) on $D_{\text{SSD}}(X)$ and $D_{\text{SSD}}(Y)$ to find the predictor matrix

$$K_{\text{SSD}} = \text{EDMD}(D_{\text{SSD}}, X, Y) = D_{\text{SSD}}(X)^\dagger D_{\text{SSD}}(Y). \tag{14}$$

The residual error $\|D_{\text{SSD}}(Y) - D_{\text{SSD}}(X)K_{\text{SSD}}\|_F$ is equal to zero and K_{SSD} fully captures the behavior of the available data. Moreover, one can replace D and K_{EDMD} in (11)-(12) by D_{SSD} and K_{SSD} to define approximated Koopman eigenfunctions and linear predictors for the dynamics. Under reasonable assumptions on the data sampling, span(D_{SSD}) is the maximal Koopman-invariant subspace in span(D) almost surely and consequently the aforementioned eigenfunctions and predictors are *exact* for *all points* (not just on the sampled data) in the state space $\mathcal{M}$ almost surely. We refer the reader to [14] for additional information about the SSD algorithm, its convergence, and its properties regarding the identification of the eigenfunctions of the Koopman operator.

3 Problem Statement

We start by noting that one can view the SSD and EDMD methods described in Section 2 as the two extreme cases in the trade-off between prediction accuracy and dictionary expressiveness. This is because, on the one hand, the SSD predictor provides almost surely exact predictions for functions in span(D_{SSD}) but, due to the pruning of the original dictionary D, this might not be sufficiently expressive to capture the system dynamics. EDMD, on the other hand, provides predictions for all functions in span(D), but there is no guarantee on the accuracy of this prediction.

Our goal is then to explore the accuracy-expressiveness spectrum in-between the extreme cases of SSD and EDMD. To do this, we seek to provide a formal data-driven characterization of how close a functional space is to being invariant under the Koopman operator (something we refer to as *invariance proximity*). Equipped with this notion, we also aim to develop computational methods that can find finite-dimensional functional spaces that meet a desired level of invariance proximity. Throughout the paper we do not assume any information about the dynamical system except the existence of a finite data set of snapshots gathered from its trajectories.

Formally, given the original dictionary D defined in (7), data snapshots matrices X, Y gathered from the dynamical system (1) defined in (6), and assuming that Assumption 2.1 holds, we seek to:

(a) provide a measure to quantify the invariance proximity of span of any dictionary $\tilde{D}$ with elements in $\mathrm{span}(D)$ solely based on available data X, Y;
(b) provide an algorithm that finds a dictionary $\tilde{D}$ with elements in $\mathrm{span}(D)$ that meets a desired level of invariance proximity;
(c) such that $\mathrm{span}(\tilde{D})$ contains the maximal Koopman-invariant subspace in $\mathrm{span}(D)$.

Requirement (c) ensures the correctness of the algorithmic solution by ensuring the maximal Koopman-invariant subspace and all Koopman eigenfunctions in $\mathrm{span}(D)$ are captured.

4 ϵ-Apart Spaces Measure Invariance Proximity

In this section we provide a quantifiable measure for invariance proximity of a subspace by studying the behavior of EDMD with respect to its dictionary. Since the true system dynamics is unknown, this measure must be based on the available data matrices X and Y. To gain a deeper understanding about the behavior of the data-dictionary pair, we offer the following interpretation of the action of the solution K_{EDMD} of the optimization (8) as a projection from $\mathcal{R}(D(Y))$ onto $\mathcal{R}(D(X))$. To see this, let $w \in \mathcal{R}(D(Y))$ be a vector of the form of $D(Y)v$. Using (9),

$$\begin{aligned} D(X)K_{\mathrm{EDMD}}v &= D(X)D(X)^{\dagger}D(Y)v \\ &= D(X)D(X)^{\dagger}w = \mathcal{P}_{D(X)}w, \end{aligned}$$

where we have used that $D(X)D(X)^{\dagger}$ is the projection operator on $\mathcal{R}(D(X))$. Using this projection viewpoint alongside (8) reveals that the residual error $\|D(Y) - D(X)K_{\mathrm{EDMD}}\|_F$ of EDMD, and consequently its accuracy on the available data, depends of how close the subspaces $\mathcal{R}(D(X))$ and $\mathcal{R}(D(Y))$ are. In fact, note that

- If D spans a Koopman-invariant subspace, we have $\mathcal{R}(D(Y)) = \mathcal{R}(D(X))$ and the residual error of EDMD is equal to zero independently of the data (as long as Assumption 2.1 holds). In this case, EDMD captures complete dynamical information about the evolution of all functions in $\mathrm{span}(D)$ and the predictor in (11) is exact;
- instead, if $\mathcal{R}(D(X)) \perp \mathcal{R}(D(Y))$, one can deduce that under Assumption 2.1, $K_{\mathrm{EDMD}} = \mathbf{0}_{\mathbf{N_d}\times\mathbf{N_d}}$ and EDMD captures no information about the dynamics. In particular, the residual error $\|D(Y) - K_{\mathrm{EDMD}}D(X)\|_F = \|D(Y)\|_F$ amounts to 100% prediction error for $D(Y)$.

These observations suggests that the proximity of the vector spaces $\mathcal{R}(D(X))$ and $\mathcal{R}(D(Y))$ can be used as a quantifiable characterization for invariance proximity of $\mathrm{span}(D)$ and consequently the prediction accuracy of EDMD. This motivates the following definition.

Definition 4.1 *(ϵ-Apart Subspaces): Given $\epsilon \geq 0$, two vector spaces $S_1, S_2 \subseteq \mathbb{R}^p$ are* ϵ-apart *if $\|\mathcal{P}_{S_1}v - \mathcal{P}_{S_2}v\|_2 \leq \epsilon\|v\|_2$, for all $v \in S_1 \cup S_2$.* □

According to this definition [2], the norm of the error induced by projecting a vector v belonging to one of the subspaces is smaller than $\epsilon\|v\|_2$. Next, we show that this notion fully characterizes equality of spaces with the case $\epsilon = 0$.

Lemma 4.2 *(0-apart Subspaces are Equal): Vector spaces $S_1, S_2 \subseteq \mathbb{R}^p$ are 0-apart if and only if $S_1 = S_2$.*

PROOF. ($\Rightarrow$): Let $v \in S_1$. By definition, $\|\mathcal{P}_{S_1}v - \mathcal{P}_{S_2}v\|_2 = \|v - \mathcal{P}_{S_2}v\|_2 = 0$, and hence $v = \mathcal{P}_{S_2}v$. Consequently, $v \in S_2$, showing $S_1 \subseteq S_2$. The inclusion $S_2 \subseteq S_1$ can be proved analogously, and we conclude $S_1 = S_2$.

($\Leftarrow$): Since $S_1 = S_2$, for all $v \in S_1 = S_2$, we have $\mathcal{P}_{S_1}v = \mathcal{P}_{S_2}v = v$. Hence, $\|\mathcal{P}_{S_1}v - \mathcal{P}_{S_2}v\|_2 = 0$, for all $v \in S_1 \cup S_2$, and the result follows. □

The next result shows that all subspaces are 1-apart.

Lemma 4.3 *(Any Two Subspaces are 1-apart): Any two vector spaces $S_1, S_2 \subseteq \mathbb{R}^p$ are 1-apart.*

PROOF. For any $v \in S_1$, one can write

$$\|\mathcal{P}_{S_1}v - \mathcal{P}_{S_2}v\|_2 = \|v - \mathcal{P}_{S_2}v\|_2 = \|(I - \mathcal{P}_{S_2})v\|_2 \leq \|v\|_2,$$

where in the last equality we have used the fact that $(I - \mathcal{P}_{S_2})$ is the projection operator on the orthogonal complement of S_2. One can write a similar argument for $v \in S_2$, which completes the proof. □

Lemmas 4.2-4.3 together imply that the range $[0, 1]$ for the parameter ϵ fully characterizes the proximity of any two subspaces. This enables us to use the concept of ϵ-apart subspaces on $D(X)$ and $D(Y)$ as a way to quantify the invariance proximity of $\mathrm{span}(D)$ under the Koopman operator associated with the system (1). Equipped with this, we reformulate next the problems (b)-(c) in Section 3.

Problem 4.4 *(Balancing Prediction Accuracy and expressiveness):Given the parameter $\epsilon \in [0, 1]$, find a dictionary $\tilde{D}$ with elements in* $\mathrm{span}(D)$ *such that*

(b) $\mathcal{R}(\tilde{D}(X))$ and $\mathcal{R}(\tilde{D}(Y))$ are ϵ-apart;
(c) $\mathrm{span}(\tilde{D})$ *contains the maximal Koopman-invariant subspace in* $\mathrm{span}(D)$. □

It is worth mentioning that

$$\epsilon^* = \min\{\epsilon \in [0, 1] \mid \mathcal{R}(D(X)), \mathcal{R}(D(Y)) \text{ are } \epsilon\text{-apart}\}$$

captures the invariance proximity, and consequently the prediction accuracy, of D. As a result, if we choose $\epsilon < \epsilon^*$ in Problem 4.4, the new dictionary would be smaller than D, leading to a decrease of the expressiveness of the resulting dictionary. Hence, the choice of parameter ϵ strikes a balance between prediction accuracy and expressiveness of the dictionary.

[2] Note that, unlike Grassmannians, e.g. [1], there is no restriction on the dimension of the subspaces in Definition 4.1.

5 Tunable Symmetric Subspace Decomposition

In this section, we design and analyze an algorithm, termed Tunable Symmetric Subspace Decomposition (T-SSD), to address Problem 4.4.

5.1 The T-SSD Algorithm

Given the dictionary D and data snapshots X, Y, the problem of finding a dictionary $\tilde{D}$ such that $\mathcal{R}(\tilde{D}(X))$ and $\mathcal{R}(\tilde{D}(Y))$ are ϵ-apart can be tackled by pruning D. We next describe informally the procedure and then formalize it in Algorithm 2.

[*Informal description*:] The pruning consists of identifying the functions that violate the desired invariance proximity condition and remove them from the dictionary. To identify such functions, we define the projection difference matrix (Step 6 in Algorithm 2)

$$G = \mathcal{P}_{D(X)} - \mathcal{P}_{D(Y)} = D(X)D(X)^{\dagger} - D(Y)D(Y)^{\dagger},$$

which is a symmetric matrix with mutually orthogonal eigenvectors spanning $\mathbb{R}^N$ (with corresponding real-valued eigenvalues). Interestingly, if all eigenvalues of G belong to $[-\epsilon, \epsilon]$, then $D(X)$ and $D(Y)$ are ϵ-apart. Otherwise, we focus our attention on the smaller subspace of $\mathbb{R}^N$ defined by

$$\mathcal{W}_{\epsilon} := \operatorname{span}\{v \in \mathbb{R}^N \mid Gv = \lambda v, \, |\lambda| \leq \epsilon\},$$

corresponding to the span of eigenvectors of G with eigenvalues in $[-\epsilon, \epsilon]$. For practical reasons, we work with a basis for $\mathcal{W}_{\epsilon}$ (Step 7 in Algorithm 2). Next, we find the largest dictionary $\tilde{D}$ with elements in $\operatorname{span}(D)$ such that $\mathcal{R}(\tilde{D}(X)), \mathcal{R}(\tilde{D}(Y)) \subseteq \mathcal{W}_{\epsilon}$ (Steps 8-9 in Algorithm 2). There are two possible outcomes:

(i) $\dim \tilde{D} = \dim D$;
(ii) $\dim \tilde{D} < \dim D$.

Scenario (i) indicates that the dictionary D does not need pruning and $\mathcal{R}(D(X)), \mathcal{R}(D(Y))$ are ϵ-apart (Steps 15-17 in Algorithm 2). On the other hand, scenario (ii) leads to a dictionary of lower dimension. However, it is not guaranteed that $\mathcal{R}(\tilde{D}(X))$ and $\mathcal{R}(\tilde{D}(Y))$ are ϵ-apart since $\tilde{D}$ is a different dictionary than D. Consequently, we re-run the process, starting with the definition of G, for the new dictionary $\tilde{D}$. This leads to an iterative implementation that stops when the dictionary cannot be reduced anymore (yielding the desired ϵ-apart subspaces).

The formalization of this procedure yields the Tunable Symmetric Subspace Decomposition (T-SSD) [3] in Algorithm 2. We make the following additional observations regarding the use of notation to provide intuition about the algorithm pseudocode: (i) we index the internal matrix variables based on the iteration number (this facilitates later the in-depth algebraic analysis); (ii) noting that, at each iteration, the dictionary elements are linear combinations of the elements of the original dictionary, we represent the dictionary at iteration i simply by a matrix C_i, which corresponds to the dictionary $D(\cdot)C_i$; (iii) using the representation in (ii), we do not need to form the dictionary and apply it on the data matrices X and Y. Instead, the effect of the dictionary at iteration i on the data can be represented as $A_i = D(X)C_i$ and $B_i = D(Y)C_i$.

Algorithm 2 Tunable Symmetric Subspace Decomposition

```
    Inputs: $D(X), D(Y) \in \mathbb{R}^{N \times N_d}$, $\epsilon \in [0,1]$
 1: Procedure T-SSD($D(X), D(Y), \epsilon$)
 2: Initialization
 3: $i \leftarrow 0$, $A_0 \leftarrow D(X)$, $B_0 \leftarrow D(Y)$, $C_0 \leftarrow I_{N_d}$
 4: while 1 do
 5:     $i \leftarrow i + 1$
 6:     $G_i \leftarrow A_{i-1}A_{i-1}^{\dagger} - B_{i-1}B_{i-1}^{\dagger}$
        ▷ projection difference
 7:     $V_i \leftarrow \operatorname{basis}(\operatorname{span}\{v \in \mathbb{R}^N \mid G_i v = \lambda v, \, |\lambda| \leq \epsilon\})$
        ▷ eigenpairs corresponding to small eigenvalues
 8:     $E_i \leftarrow$ Symmetric-Intersection($V_i, A_{i-1}, B_{i-1}$)
        ▷ Find largest dictionary matrices in $V_i$ (Algorithm 3)
 9:     $C_i \leftarrow C_{i-1}E_i$                    ▷ reduce subspace
10:     $A_i \leftarrow A_{i-1}E_i$, $B_i \leftarrow B_{i-1}E_i$
        ▷ calculate new dictionary matrices
11:     if $E_i = 0$ then
12:         return 0
        ▷ subspace does not exist, returning scalar 0
13:         break
14:     end if
15:     if $\sharp\text{rows}(E_i) \leq \sharp\text{cols}(E_i)$ then
16:         return $C_i$                 ▷ procedure is complete
17:         break
18:     end if
19: end while
```

Algorithm 3 describes the Symmetric-Intersection function in Step 8 of T-SSD: this strategy corresponds to the computation described above of the largest dictionary $\tilde{D}$ such that $\mathcal{R}(\tilde{D}(X))$ and $\mathcal{R}(\tilde{D}(Y))$ belong to the reduced subspace W_{ϵ}. Similarly to Algorithm 2, instead of actually forming the reduced dictionary, Algorithm 3 uses the matrix-based representation of the dictionary. Next, we explain the steps of the algorithm and the reason behind its naming. Given input matrices V, A, and B, Step 6 in Algorithm 3 identifies W_A such that $\mathcal{R}(AW_A) = \mathcal{R}(V) \cap \mathcal{R}(A)$ (see Lemma A.1). Then, again in Step 13 the algorithm (by virtue of Lemma A.1) finds the matrix Z_B such that $\mathcal{R}(BW_AZ_B) = \mathcal{R}(V) \cap \mathcal{R}(BW_A)$. The output matrix $E := \operatorname{basis}(W_AZ_B)$ (cf. Step 13) then specifies the largest subspaces $\mathcal{R}(AE)$, $\mathcal{R}(BE)$ both belonging to $\mathcal{R}(V)$. Note the symmetry in this specification: if a linear combination of the columns of A is in $\mathcal{R}(V)$, then the same linear combination of columns of B belongs to $\mathcal{R}(V)$. Moreover, Algorithm 3 breaks and returns 0 if any of the aforementioned intersections only contain the zero vector (Steps 2-4 and Steps 7-9).

Remark 5.1 *(Implementation of Algorithm 3 on Finite-Precision Computers):* The accuracy of the implementation of Algorithm 3 depends on the calculation of the null space of several matrices, which might be sensitive to round-off errors. To circumvent this issue, one can set sufficiently small (according to a desired accuracy level) singular values of the matrices to zero. □

5.2 Basic Properties of T-SSD

Our end goal now is to show that the T-SSD algorithm solves Problem 4.4 and unveil its relationship with the EDMD and SSD methods. In order to do so, we establish here several

[3] In Algorithms 2-3, the outputs of null(A) and basis(A) are matrices whose columns form orthonormal bases for the null space of A and $\mathcal{R}(A)$, respectively.

Algorithm 3 Symmetric Intersection

```
    Inputs: $V \in \mathbb{R}^{n\times m}$ and $A, B \in \mathbb{R}^{n\times p}$
 1: Procedure Symmetric-Intersection($V, A, B$)
 2: if null($[V, A]$) $= \emptyset$ then
 3:     return 0
 4:     break
 5: else
 6:     $\begin{bmatrix} W_V \\ W_A \end{bmatrix} \leftarrow$ null($[V, A]$)
        ▷ $\sharp$cols($V$) $= \sharp$rows($W_V$), $\sharp$cols($A$) $= \sharp$rows($W_A$)
 7:     if null($[V, BW_A]$) $= \emptyset$ then
 8:         return 0
 9:         break
10:     end if
11:     $\begin{bmatrix} Z_V \\ Z_B \end{bmatrix} \leftarrow$ null($[V, BW_A]$)
        ▷ $\sharp$cols($V$) $= \sharp$rows($Z_V$), $\sharp$cols($BW_A$) $= \sharp$rows($Z_B$)
12: end if
13: return basis($W_A Z_B$)   ▷ Returning an orthogonal basis
```

basic algorithm properties.

Proposition 5.2 *(Properties of* Symmetric-Intersection*): Let matrices V, A, B have full column rank and $E =$* Symmetric-Intersection(V, A, B)*. Then,*

(a) $E = 0$ or $E^T E = I$;
(b) $\mathcal{R}(AE), \mathcal{R}(BE) \subseteq \mathcal{R}(V)$;
(c) E is maximal, i.e., any nonzero matrix F such that $\mathcal{R}(AF), \mathcal{R}(BF) \subseteq \mathcal{R}(V)$ satisfies $\mathcal{R}(F) \subseteq \mathcal{R}(E)$.

PROOF. (a) There are three ways for Algorithm 3 to terminate. If the algorithm executes Steps 2-4 or Steps 7-9, we have $E = 0$ by definition. Otherwise, the algorithm executes Step 13. Hence, noting that W_A and Z_B exist and the basis function returns an orthonormal basis for $W_A Z_B$, one can conclude $E^T E = I$.

(b) The case $E = 0$ is trivial. Suppose that $E \neq 0$ and hence has full column rank according to part (a). By definition, $\mathcal{R}(E) = \mathcal{R}(W_A Z_B)$. Consequently, based on Step 11 of the algorithm and using Lemma A.1, we deduce

$$\begin{aligned}\mathcal{R}(BE) &= \mathcal{R}(BW_A Z_B) \\ &= \mathcal{R}(BW_A) \cap \mathcal{R}(V) \subseteq \mathcal{R}(V),\end{aligned} \tag{15}$$

where in the first equality, we used Lemma A.2. Moreover, from the definition of E, one can deduce that $\mathcal{R}(E) \subseteq \mathcal{R}(W_A)$. In addition, based on Lemma A.2, we have $\mathcal{R}(AE) \subseteq \mathcal{R}(AW_A)$. Using the previous inequality in conjunction with Lemma A.1 applied to Step 6 of the algorithm, one can write

$$\mathcal{R}(AE) \subseteq \mathcal{R}(AW_A) = \mathcal{R}(A) \cap \mathcal{R}(V) \subseteq \mathcal{R}(V), \tag{16}$$

which in conjunction with (15) concludes the proof of (b).

(c) Without loss of generality, we assume that F has full column rank (if that is not the case, one can consider another matrix $\bar{F}$ with full column rank such that $\mathcal{R}(F) = \mathcal{R}(\bar{F})$). Since $\mathcal{R}(AF) \subseteq \mathcal{R}(V)$, we have $\mathcal{R}(AF) \subseteq \mathcal{R}(A) \cap \mathcal{R}(V)$, which leads to $\mathcal{R}(AF) \subseteq \mathcal{R}(AW_A)$ based on (16). Moreover, one can use Lemma A.2 to deduce that $\mathcal{R}(F) \subseteq \mathcal{R}(W_A)$. Since F and W_A both have full column rank, there exists F_W with full column rank such that

$$F = W_A F_W. \tag{17}$$

Considering that $\mathcal{R}(BF) \subseteq \mathcal{R}(V)$ and $\mathcal{R}(BW_A F_W) \subseteq \mathcal{R}(BW_A)$ in combination with (17), we deduce $\mathcal{R}(BF) = \mathcal{R}(BW_A F_W) \subseteq \mathcal{R}(BW_A) \cap \mathcal{R}(V) = \mathcal{R}(BW_A Z_B)$, where the last equality follows from (15). Based on Lemma A.2, we deduce $\mathcal{R}(F) \subseteq \mathcal{R}(W_A Z_B) = \mathcal{R}(E)$. □

Next, we show that T-SSD terminates after a finite number of iterations.

Proposition 5.3 *(Finite-time Termination of T-SSD Algorithm): The T-SSD algorithm terminates after at most N_d iterations.*

PROOF. We reason by contradiction. Suppose that the algorithm does not terminate before iteration $N_d + 1$. Hence, the algorithm does not execute Steps 12-13 or Steps 16-17 in the first N_d iterations. Therefore, the conditions in Steps 11 and 15 do not hold. Using Proposition 5.2(a), one can write

$$\sharp\text{rows}(E_i) > \sharp\text{rows}(E_i) - 1 \geq \sharp\text{cols}(E_i), \tag{18}$$

for all $i \in \{1, \ldots, N_d\}$. In addition, based on the definition of the E_i's, one can deduce $\sharp\text{cols}(E_i) = \sharp\text{rows}(E_{i+1})$, for all $i \in \{1, \ldots, N_d\}$. Combining this with (18) leads to $\sharp\text{rows}(E_1) \geq \sharp\text{cols}(E_{N_d}) + N_d$. This fact together with $\sharp\text{rows}(E_1) = N_d$ and $\sharp\text{cols}(E_{N_d}) = \sharp\text{cols}(C_{N_d})$ (cf. Step 9) implies that $\sharp\text{cols}(C_{N_d}) \leq 0$, contradicting $\sharp\text{cols}(C_{N_d}) \geq 1$. □

Next, we study basic properties of the internal matrices of the T-SSD algorithm.

Lemma 5.4 *(Properties of T-SSD Matrices): Let the T-SSD algorithm terminate in L time steps. Then,*

(a) $\forall i \in \{0, \ldots, L-1\}$, $\mathcal{R}(C_{i+1}) \subseteq \mathcal{R}(C_i)$;
(b) $\forall i \in \{0, \ldots, L-1\}$, $C_i^T C_i = I$;
(c) $C_L = \mathbf{0}$ or $C_L^T C_L = I$,

where C_i denotes T-SSD's ith internal matrix, cf. Algorithm 2.

PROOF. (a) According to Step 9 of the algorithm, $C_{i+1} = C_i E_{i+1}$. Hence, $\mathcal{R}(C_{i+1}) = \mathcal{R}(C_i E_{i+1}) \subseteq \mathcal{R}(C_i)$.

(b) For $i = 0$, the result holds by definition. Moreover, since the algorithm does not terminate until iteration L, it does not execute Steps 12-13 in iterations $\{1, \ldots, L-1\}$. Hence, $E_i \neq 0$ and based on Proposition 5.2(a), we have

$$E_i^T E_i = I, \forall i \in \{1, \ldots, L-1\}. \tag{19}$$

Moreover, from Step 9, $C_i = C_0 E_1 E_2 \cdots E_i$, $\forall i \in \{1, \ldots, L-1\}$. This in conjunction with (19) and $C_0 = I_{N_d}$, implies $C_i^T C_i = I$ for all $i \in \{1, \ldots, L-1\}$, as claimed.

(c) Note that $C_L = C_{L-1} E_L$. Based on Proposition 5.2(a), either $E_L = 0$ or $E_L^T E_L = I$. In the former case, we have $C_L = \mathbf{0}$. In the latter case, $C_L^T C_L = C_{L-1}^T E_L^T E_L C_{L-1} = C_{L-1}^T C_{L-1} = I$, where in the last equality we used (b). □

For convenience, let

$$C_{\text{T-SSD}} := \text{T-SSD}(D(X), D(Y), \epsilon), \tag{20}$$

denote the output of the T-SSD algorithm. This leads to the definition of the T-SSD dictionary

$$D_{\text{T-SSD}}(\cdot) := D(\cdot)C_{\text{T-SSD}}. \tag{21}$$

To extract the dynamical information associated with the Koopman operator on $\text{span}(D_{\text{T-SSD}})$, we use EDMD. According to (9), we find the T-SSD prediction matrix as

$$K_{\text{T-SSD}} := \text{EDMD}(D_{\text{T-SSD}}, X, Y) = D_{\text{T-SSD}}(X)^{\dagger} D_{\text{T-SSD}}(Y). \tag{22}$$

We can also define approximated Koopman eigenfunctions according to (12) using the eigendecomposition of $K_{\text{T-SSD}}$ and the dictionary $D_{\text{T-SSD}}$. In addition, following (11), given any function $f \in \text{span}(D_{\text{T-SSD}})$ described by $f(\cdot) = D_{\text{T-SSD}}(\cdot)w$, we can define the T-SSD predictor of $\mathcal{K}f$ on $\text{span}(D_{\text{T-SSD}})$ as

$$\mathfrak{P}^{\text{T-SSD}}_{\mathcal{K}f}(\cdot) = D_{\text{T-SSD}}(\cdot) K_{\text{T-SSD}} w. \tag{23}$$

Remark 5.5 *(Computational Complexity of T-SSD):* Given N data snapshots and N_d dictionary functions, and considering the complexity of scalar operations as $O(1)$, the most time-consuming step of Algorithm 2 is Step 7, which requires the eigendecomposition of an $N \times N$ matrix and takes $O(N^3)$ operations. Based on Proposition 5.3, the algorithm terminates after at most N_d iterations, resulting in a total complexity of $O(N^3 N_d)$. □

6 T-SSD Balances Accuracy and Expressiveness

In this section we show that the output of T-SSD balances prediction accuracy and expressiveness as prescribed by the design parameter $\epsilon \in [0, 1]$.

6.1 T-SSD Identifies ϵ-Apart Subspaces

Here, we show that T-SSD solves Problem 4.4(b).

Theorem 6.1 *(T-SSD Output Subspaces are ϵ-Apart): $\mathcal{R}(D_{\text{T-SSD}}(X))$ and $\mathcal{R}(D_{\text{T-SSD}}(Y))$ are ϵ-apart.*

PROOF. Let $L \le N_d$ be the number of iterations for convergence of the T-SSD algorithm (cf. Proposition 5.3). Based on Proposition 5.2(a), we have $E_L = 0$ or $E_L^T E_L = I$. With the notation of Algorithm 2, in the former case, the algorithm executes Steps 12-13 at iteration L and consequently $C_{\text{T-SSD}} = 0$. Therefore,

$$\mathcal{R}(D_{\text{T-SSD}}(X)) = \mathcal{R}(D_{\text{T-SSD}}(Y)) = \{\mathbf{0}_{N\times 1}\},$$

and the result holds trivially. Now, suppose that $E_L^T E_L = I$. Hence, E_L has full column rank and consequently $\sharp\text{rows}(E_L) \ge \sharp\text{cols}(E_L)$. However, since the algorithm executes Steps 16-17, the condition in Step 15 holds and one can write $\sharp\text{rows}(E_L) = \sharp\text{cols}(E_L)$. Therefore, since E_L has full column rank, it is a nonsingular square matrix and

$$\mathcal{R}(C_L) = \mathcal{R}(C_{L-1}E_L) = \mathcal{R}(C_{L-1}), \tag{24a}$$
$$\mathcal{R}(A_L) = \mathcal{R}(A_{L-1}E_L) = \mathcal{R}(A_{L-1}), \tag{24b}$$
$$\mathcal{R}(B_L) = \mathcal{R}(B_{L-1}E_L) = \mathcal{R}(B_{L-1}). \tag{24c}$$

At iteration L, one can use Steps 6 and 7 in conjunction with the fact that the eigenvectors of G_L are mutually orthogonal to write

$$\begin{aligned}\|G_L v\|_2 &= \|A_{L-1}A_{L-1}^{\dagger}v - B_{L-1}B_{L-1}^{\dagger}v\|_2 \\ &= \|\mathcal{P}_{A_{L-1}}v - \mathcal{P}_{B_{L-1}}v\|_2 \le \epsilon\|v\|_2,\end{aligned} \tag{25}$$

for all $v \in \mathcal{R}(V_L)$. Moreover, based on definition of E_L and Proposition 5.2(b),

$$\mathcal{R}(A_{L-1}E_L), \mathcal{R}(B_{L-1}E_L) \subseteq \mathcal{R}(V_L). \tag{26}$$

Consequently, using $A_L = D(X)C_L$ and $B_L = D(Y)C_L$, and equations (24)-(26), we deduce

$$\begin{aligned}\|\mathcal{P}_{D(X)C_L}v - \mathcal{P}_{D(Y)C_L}v\|_2 &= \|\mathcal{P}_{A_L}v - \mathcal{P}_{B_L}v\|_2 \\ &= \|\mathcal{P}_{A_{L-1}}v - \mathcal{P}_{B_{L-1}}v\|_2 \le \epsilon\|v\|_2,\end{aligned}$$

for all $v \in \mathcal{R}(D(X)C_L) \cup \mathcal{R}(D(Y)C_L)$. Since $C_L = C_{\text{T-SSD}}$, and given the definition (21) of the T-SSD dictionary, this can be rewritten as $\|\mathcal{P}_{D_{\text{T-SSD}}(X)}v - \mathcal{P}_{D_{\text{T-SSD}}(Y)}v\|_2 \le \epsilon\|v\|_2$, for all $v \in \mathcal{R}(D_{\text{T-SSD}}(X)) \cup \mathcal{R}(D_{\text{T-SSD}}(Y))$. Hence, $\mathcal{R}(D_{\text{T-SSD}}(X))$ and $\mathcal{R}(D_{\text{T-SSD}}(Y))$ are ϵ-apart. □

We next build on Theorem 6.1 to characterize the accuracy of predictions (23) for any function in $\text{span}(D_{\text{T-SSD}})$ on the available data.

Theorem 6.2 *(Relative Root Mean Square Error (RRMSE) of Koopman Predictions by T-SSD are Bounded by ϵ): For any function $f \in \text{span}(D_{\text{T-SSD}})$,*

$$\frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N} |\mathcal{K}f(x_i) - \mathfrak{P}^{\text{T-SSD}}_{\mathcal{K}f}(x_i)|^2}}{\sqrt{\frac{1}{N}\sum_{i=1}^{N} |\mathcal{K}f(x_i)|^2}} \le \epsilon \tag{27}$$

where x_i^T is the ith row of X and $\mathfrak{P}^{\text{T-SSD}}_{\mathcal{K}f}$ is defined in (23).

PROOF. For convenience, we use the compact notation $\tilde{D}$ to refer to $D_{\text{T-SSD}}$ throughout the proof. We first prove the statement for real-valued functions in $\text{span}(\tilde{D})$. Let $f(\cdot) = \tilde{D}(\cdot)w$ with $w \in \mathbb{R}^{\sharp\text{cols}(C_{\text{T-SSD}})}$. From Theorem 6.1, one can write $\|(\mathcal{P}_{\tilde{D}(Y)} - \mathcal{P}_{\tilde{D}(X)})v\|_2 \le \epsilon\|v\|_2$, for all $v \in \mathcal{R}(\tilde{D}(X)) \cup \mathcal{R}(\tilde{D}(Y))$. One can rewrite this equation as

$$\|(\tilde{D}(Y)\tilde{D}(Y)^{\dagger} - \tilde{D}(X)\tilde{D}(X)^{\dagger})v\|_2 \le \epsilon\|v\|_2, \tag{28}$$

for all $v \in \mathcal{R}(\tilde{D}(X)) \cup \mathcal{R}(\tilde{D}(Y))$. In addition, using equations (22) and (23), and the fact that $\mathcal{K}f(x_i) = f \circ T(x_i) = f(y_i)$ for all $i \in \{1, \ldots, N\}$, one can write

$$\begin{aligned}\sqrt{\sum_{i=1}^{N} \left|\mathcal{K}f(x_i) - \mathfrak{P}^{\text{T-SSD}}_{\mathcal{K}f}(x_i)\right|^2} &= \|(\tilde{D}(Y) - \tilde{D}(X)K_{\text{T-SSD}})w\|_2 \\ &= \|(\tilde{D}(Y) - \tilde{D}(X)\tilde{D}(X)^{\dagger}\tilde{D}(Y))w\|_2 \\ &= \|\big(\tilde{D}(Y)\tilde{D}(Y)^{\dagger} - \tilde{D}(X)\tilde{D}(X)^{\dagger}\big)\tilde{D}(Y)w\|_2,\end{aligned}$$

where we have used $\tilde{D}(Y) = \tilde{D}(Y)\tilde{D}(Y)^\dagger \tilde{D}(Y)$ in the last equality. Moreover, since $\tilde{D}(Y)w \in \mathcal{R}(\tilde{D}(X)) \cup \mathcal{R}(\tilde{D}(Y))$, one can use this equation in conjunction with (28) to write

$$\sqrt{\sum_{i=1}^{N} \left|\mathcal{K}f(x_i) - \mathfrak{P}^{\text{T-SSD}}_{\mathcal{K}f}(x_i)\right|^2} \leq \epsilon \|\tilde{D}(Y)w\|_2 = \epsilon \sqrt{\sum_{i=1}^{N} |\mathcal{K}f(x_i)|^2}.$$

Scaling both sides by $N^{-\frac{1}{2}}$ yields (27) for real-valued functions in $\text{span}(\tilde{D})$.

For the complex-valued case, let $f(\cdot) = \tilde{D}(\cdot)w$ with $w = w_{\text{re}} + jw_{\text{im}}$, $w_{\text{re}}, w_{\text{im}} \in \mathbb{R}^{\sharp\text{cols}(C_{\text{T-SSD}})}$ and $w_{\text{im}} \neq 0$.

Consider the decompositions of f and $\mathfrak{P}^{\text{T-SSD}}_{\mathcal{K}f}$ as $f(\cdot) = f_{\text{re}}(\cdot) + jf_{\text{im}}(\cdot)$ and $\mathfrak{P}^{\text{T-SSD}}_{\mathcal{K}f}(\cdot) = \mathfrak{P}^{\text{T-SSD}}_{\mathcal{K}f_{\text{re}}}(\cdot) + j\mathfrak{P}^{\text{T-SSD}}_{\mathcal{K}f_{\text{im}}}(\cdot)$, where

$$\begin{aligned} &f_{\text{re}}(\cdot) = \tilde{D}(\cdot)w_{\text{re}}, && f_{\text{im}}(\cdot) = \tilde{D}(\cdot)w_{\text{im}}, \\ &\mathfrak{P}^{\text{T-SSD}}_{\mathcal{K}f_{\text{re}}}(\cdot) = \tilde{D}(\cdot)K_{\text{T-SSD}}w_{\text{re}}, && \mathfrak{P}^{\text{T-SSD}}_{\mathcal{K}f_{\text{im}}}(\cdot) = \tilde{D}(\cdot)K_{\text{T-SSD}}w_{\text{im}}. \end{aligned} \tag{29}$$

Using (27) for the real-valued functions in (29),

$$\begin{aligned} \sum_{i=1}^{N} |\mathcal{K}f_{\text{re}}(x_i) - \mathfrak{P}^{\text{T-SSD}}_{\mathcal{K}f_{\text{re}}}(x_i)|^2 &\leq \epsilon^2 \sum_{i=1}^{N} |\mathcal{K}f_{\text{re}}(x_i)|^2, \\ \sum_{i=1}^{N} |\mathcal{K}f_{\text{im}}(x_i) - \mathfrak{P}^{\text{T-SSD}}_{\mathcal{K}f_{\text{im}}}(x_i)|^2 &\leq \epsilon^2 \sum_{i=1}^{N} |\mathcal{K}f_{\text{im}}(x_i)|^2. \end{aligned}$$

By adding these two inequalities, using (29), and noting that $|g|^2 = |g_{\text{re}}|^2 + |g_{\text{im}}|^2$ for $g = g_{\text{re}} + jg_{\text{im}}$, one can write

$$\sum_{i=1}^{N} |\mathcal{K}f(x_i) - \mathfrak{P}^{\text{T-SSD}}_{\mathcal{K}f}(x_i)|^2 \leq \epsilon^2 \sum_{i=1}^{N} |\mathcal{K}f(x_i)|^2,$$

and (27) follows. □

Theorem 6.2 ensures that each member of the vector space of functions identified by T-SSD has prediction error bounded by the accuracy parameter ϵ.

Remark 6.3 *(T-SSD Bounds the Relative L_2-norm Error of Koopman Predictions under Empirical Measure by ϵ):* Given the functions in $\text{span}(D)$ and their composition with T are measurable, consider the empirical measure $\mu = \frac{1}{N}\sum_{k=1}^{N} \delta_{x_k}$, where δ_{x_k} denotes the Dirac delta function at x_k, the kth row of X. Then Theorem 6.2 can be rewritten as

$$\frac{\|\mathcal{K}f - \mathfrak{P}^{\text{T-SSD}}_{\mathcal{K}f}\|_{L_2}}{\|\mathcal{K}f\|_{L_2}} \leq \epsilon, \; \forall f \in \text{span}(D_{\text{T-SSD}}),$$

where the L_2-norm is calculated with respect to the empirical measure μ. □

6.2 T-SSD Captures Maximal Koopman-Invariant Subspace

Here, we show that T-SSD also solves Problem 4.4(c). To do this, we study the relationship of the algorithm with Koopman eigenfunctions and invariant subspaces. We first show that the T-SSD matrices capture the maximal Koopman-invariant subspaces in the span of the original dictionary D.

Theorem 6.4 *(T-SSD Matrices Capture the Maximal Koopman-Invariant Subspace): Let $\mathcal{I}_{\max}$ denote the maximal Koopman-invariant subspace in* $\text{span}(D)$ *and let $C_{\max}$ be a full-column rank matrix such that $D(\cdot)C_{\max}$ spans $\mathcal{I}_{\max}$ (if $\mathcal{I}_{\max} = \{0\}$, we set $C_{\max} = 0$). Then, for any $\epsilon \in [0,1]$,*

$$\mathcal{R}(C_{\max}) \subseteq \mathcal{R}(C_i), \quad \forall i \in \{0, \ldots, L\},$$

where L and C_i denote, respectively, the termination step and the ith internal matrix of T-SSD.

PROOF. The result holds trivially if $\mathcal{I}_{\max} = \{0\}$. For the case $\mathcal{I}_{\max} \neq \{0\}$, we reason by induction. For $i = 0$, columns of C_0 span the whole space. Hence, $\mathcal{R}(C_{\max}) \subseteq \mathcal{R}(C_0)$. Next, assume $\mathcal{R}(C_{\max}) \subseteq \mathcal{R}(C_i)$ for $i \in \{0, 1, \ldots, L-1\}$ and let us prove $\mathcal{R}(C_{\max}) \subseteq \mathcal{R}(C_{i+1})$. The invariance of $\mathcal{I}_{\max}$ implies that $\mathcal{R}(D(X)C_{\max}) = \mathcal{R}(D(Y)C_{\max})$. Using the definition of matrices A_0, B_0 in Algorithm 2, this can be equivalently written as $\mathcal{R}(A_0C_{\max}) = \mathcal{R}(B_0C_{\max})$. Since $\mathcal{R}(C_{\max}) \subseteq \mathcal{R}(C_i)$, using Lemma A.2, we deduce

$$\mathcal{R}(A_0C_{\max}) \subseteq \mathcal{R}(A_0C_i), \quad \mathcal{R}(B_0C_{\max}) \subseteq \mathcal{R}(B_0C_i).$$

Hence, $\mathcal{P}_{A_0C_i}w = w = \mathcal{P}_{B_0C_i}w$, for all $w \in \mathcal{R}(A_0C_{\max}) = \mathcal{R}(B_0C_{\max})$, or equivalently,

$$\|\mathcal{P}_{A_0C_i}w - \mathcal{P}_{B_0C_i}w\|_2 = 0, \quad \forall w \in \mathcal{R}(A_0C_{\max}) = \mathcal{R}(B_0C_{\max}). \tag{30}$$

Now, noting that $A_i = A_0C_i$ and $B_i = B_0C_i$, one can use Step 6 of Algorithm 2 and write $G_{i+1}v = \mathcal{P}_{A_0C_i}v - \mathcal{P}_{B_0C_i}v$, for all $v \in \mathbb{R}^N$. This, combined with (30), yields $\mathcal{R}(A_0C_{\max}) = \mathcal{R}(B_0C_{\max}) \subseteq \text{null}(G_{i+1})$. Therefore, since the eigenvectors of G_{i+1} with zero eigenvalue span $\text{null}(G_{i+1})$, we deduce from Step 7,

$$\mathcal{R}(A_0C_{\max}) = \mathcal{R}(B_0C_{\max}) \subseteq \mathcal{R}(V_{i+1}). \tag{31}$$

Based on the induction hypothesis $\mathcal{R}(C_{\max}) \subset \mathcal{R}(C_i)$, and noting that $C_{\max}$ and C_i have full column rank ($C_{\max}$ by definition and C_i from Lemma 5.4(b)), there exits a full-column rank matrix F_i such that

$$C_{\max} = C_iF_i. \tag{32}$$

Now, using (31)-(32), in conjunction with Proposition 5.2(c), we deduce $\mathcal{R}(F_i) \subseteq \mathcal{R}(E_{i+1})$. Consequently, one can use Lemma A.2 and write $\mathcal{R}(C_{\max}) = \mathcal{R}(C_iF_i) \subseteq \mathcal{R}(C_iE_{i+1}) = \mathcal{R}(C_{i+1})$, concluding the proof. □

Theorem 6.4 implies that the subspace identified by T-SSD contains the maximal Koopman-invariant subspace in $\text{span}(D)$.

Corollary 6.5 *(T-SSD Subspace Contains the Maximal Koopman-Invariant Subspace): Let $\mathcal{I}_{\max}$ be the maximal*

Koopman-invariant subspace in span(D). *Given* $\epsilon \in [0,1]$, *let* $C_{\text{T-SSD}}$ *and* $D_{\text{T-SSD}}$ *be the output and dictionary identified by T-SSD according to* (20)-(21). *Then,* $\mathcal{I}_{\max} \subseteq \text{span}(D_{\text{T-SSD}})$.

The next result shows that the eigendecomposition of $K_{\text{T-SSD}}$ captures all Koopman eigenfunctions (and corresponding eigenvalues) in the span of the original dictionary.

Proposition 6.6 *($K_{\text{T-SSD}}$ Captures All Koopman Eigenfunctions in* span(D)*): Let* ϕ *be a Koopman eigenfunction in* span(D) *with eigenvalue* λ. *For* $\epsilon \in [0,1]$, *let* $K_{\text{T-SSD}}$ *in* (22) *be the T-SSD predictor matrix. Then,* $\phi \in \text{span}(D_{\text{T-SSD}})$ *and there exists* w *with* $K_{\text{T-SSD}}w = \lambda w$ *such that* $\phi(\cdot) = D_{\text{T-SSD}}(\cdot)w$.

PROOF. Note that ϕ must belong to the maximal Koopman-invariant subspace $\mathcal{I}_{\max}$ in span(D) which, from Corollary 6.5, is included in $\text{span}(D_{\text{T-SSD}}) = \text{span}(D(\cdot)C_{\text{T-SSD}})$. Therefore, there exists a complex vector w of appropriate size such that $\phi(\cdot) = D_{\text{T-SSD}}(\cdot)w$. Using now the interpretation of $K_{\text{T-SSD}}$ as the EDMD solution with dictionary $D_{\text{T-SSD}}$ and data X, Y, it follows from [14, Lemma 4.1](b) that $K_{\text{T-SSD}}w = \lambda w$, as claimed. □

Proposition 6.6 states that all eigenfunctions in the span of the original dictionary D belong to the set of approximated eigenfunctions calculated with the dictionary $D_{\text{T-SSD}}$ defined by T-SSD.

Remark 6.7 *(Monotonicity of T-SSD Subspaces):* In general, the output of the T-SSD algorithm is not monotonic as a function of the design parameter ϵ, i.e., it might be the case that $\text{span}(D^{\epsilon_1}_{\text{T-SSD}}) \not\subseteq \text{span}(D^{\epsilon_2}_{\text{T-SSD}})$ for $\epsilon_1 < \epsilon_2$. In case monotonicity is desirable for a specific application, one can modify Step 7 of Algorithm 2 to remove only the eigenvector with the largest eigenvalue (in magnitude) that exceeds the desired accuracy level. This modification ensures monotonicity in ϵ at the cost of requiring the modified algorithm more iterations to terminate. All the results remain valid for the modified version of the algorithm. □

7 EDMD and SSD are Special Cases of T-SSD

Consistent with our assertion that T-SSD balances accuracy and expressiveness, here we show that EDMD on the original dictionary (maximum expressiveness) corresponds to T-SSD with $\epsilon = 1$ and that SSD (maximum accuracy) corresponds to T-SSD with $\epsilon = 0$ [4]. We start by showing an important property of EDMD.

Lemma 7.1 *(Linear Transformations Do not Change the Information Extracted by EDMD): Let* D_1 *and* D_2 *be two dictionaries such that* $D_1(\cdot) = D_2(\cdot)R$, *with* R *invertible. Let Assumption 2.1 hold for both dictionaries given data matrices* X *and* Y. *Define*

$$K^1_{\text{EDMD}} = \text{EDMD}(D_1, X, Y) = D_1(X)^\dagger D_1(Y),$$
$$K^2_{\text{EDMD}} = \text{EDMD}(D_2, X, Y) = D_2(X)^\dagger D_2(Y).$$

Then, $K^1_{\text{EDMD}} = R^{-1}K^2_{\text{EDMD}}R$. *Therefore,* (λ, v) *is an eigenpair of* K^1_{EDMD} *if and only if* (λ, Rv) *is an eigenpair of* K^2_{EDMD}.

[4] We refer to K_{SSD} and $K_{\text{T-SSD}}$ as SSD and T-SSD Koopman approximations, which can be calculated by applying EDMD on dictionaries identified by SSD and T-SSD respectively.

PROOF. Based on Assumption 2.1, we have $K^1_{\text{EDMD}} = \big(D_1(X)^T D_1(X)\big)^{-1} D_1(X)^T D_1(Y)$. Using $D_1(\cdot) = D_2(\cdot)R$,

$$\begin{aligned} K^1_{\text{EDMD}} &= \big(R^T D_2(X)^T D_2(X) R\big)^{-1} R^T D_2(X)^T D_2(Y) R \\ &= R^{-1}\big(D_2(X)^T D_2(X)\big)^{-1} D_2(X)^T D_2(Y) R \\ &= R^{-1} D_2(X)^\dagger D_2(Y) R = R^{-1} K^2_{\text{EDMD}} R. \end{aligned}$$

The rest follows from the properties of similarity transformations. □

Lemma 7.1 states that the dynamical information captured by the EDMD algorithm remains the same under linear transformation of the dictionary. Note that the result does not require the dictionaries to span a Koopman-invariant subspace. We are ready to show that EDMD applied to the original dictionary is a special case of T-SSD.

Theorem 7.2 *(EDMD is a Special Case of T-SSD with* $\epsilon = 1$*): For* $\epsilon = 1$, *let* $D_{\text{T-SSD}}$ *be the dictionary identified by T-SSD, cf.* (21). *Then,* $\text{span}(D_{\text{T-SSD}}) = \text{span}(D)$, *and* $K_{\text{T-SSD}} = \text{EDMD}(D_{\text{T-SSD}}, X, Y)$ *and* $K_{\text{EDMD}} = \text{EDMD}(D, X, Y)$ *are similar and capture the same dynamical information.*

PROOF. In the first iteration of Algorithm 2, one can use Step 6 and the definition of A_0 and B_0 to write

$$G_1 = A_0 A_0^\dagger - B_0 B_0^\dagger = \mathcal{P}_{D(X)} - \mathcal{P}_{D(Y)}.$$

Since G_1 is symmetric, its eigenvalues are real. Moreover, they belong to $[-1, 1]$, see e.g. [2, Lemma 1]. Therefore, since $\epsilon = 1$, using Step 7, one can deduce that the columns of V_1 span $\mathbb{R}^N$. As a result,

$$\mathcal{R}(D(X)) = \mathcal{R}(A_0) = \mathcal{R}(A_0 I_{N_d}) \subseteq \mathcal{R}(V_1) = \mathbb{R}^N,$$
$$\mathcal{R}(D(Y)) = \mathcal{R}(B_0) = \mathcal{R}(B_0 I_{N_d}) \subseteq \mathcal{R}(V_1) = \mathbb{R}^N.$$

This, combined with the maximality of E_1 defined in Step 8, cf. Proposition 5.2(c), implies $\mathcal{R}(I_{N_d}) \subseteq \mathcal{R}(E_1)$. Hence, E_1 is nonzero and has full column rank (cf. Proposition 5.2(a)). As a result, nothing that $\sharp\text{rows}(E_1) = N_d$, we deduce that E_1 is a nonsingular square matrix. Therefore, $\mathcal{R}(C_1) = \mathcal{R}(C_0 E_1) = \mathbb{R}^{N_d}$. This and the fact that E_1 is square mean that the condition in Step 15 is met and the algorithm executes Steps 16-17. Consequently, $C_{\text{T-SSD}} = C_1$ is a nonsingular square matrix and $\text{span}(D_{\text{T-SSD}}(\cdot)) = \text{span}(D(\cdot)C_{\text{T-SSD}}) = \text{span}(D(\cdot))$, so $D_{\text{T-SSD}}$ is a (potentially different) basis for the space spanned by D. The rest of the statement follows from Lemma 7.1. □

The SSD algorithm is also a special case of T-SSD.

Theorem 7.3 *(SSD is a Special Case of T-SSD with* $\epsilon = 0$*): Let* $D_{\text{SSD}}(\cdot)$ *be the dictionary identified by SSD, cf.* (13)*, and, for* $\epsilon = 0$, *let* $D_{\text{T-SSD}}$ *be the dictionary identified by T-SSD, cf.* (21). *Then,* $\text{span}(D_{\text{T-SSD}}) = \text{span}(D_{\text{SSD}})$, *and* $K_{\text{T-SSD}} = \text{EDMD}(D_{\text{T-SSD}}, X, Y)$ *and* $K_{\text{SSD}} = \text{EDMD}(D_{\text{SSD}}, X, Y)$ *are similar and capture the same dynamical information.*

PROOF. Since $\epsilon = 0$, Theorem 6.1 implies that $\mathcal{R}(D(X)C_{\text{T-SSD}})$ and $\mathcal{R}(D(Y)C_{\text{T-SSD}})$ are 0-apart. Therefore, from Lemma 4.2,

$\mathcal{R}(D(X)C_{\text{T-SSD}}) = \mathcal{R}(D(X)C_{\text{T-SSD}})$. This, together with Theorem 2.2(c), implies

$$\mathcal{R}(C_{\text{T-SSD}}) \subseteq \mathcal{R}(C_{\text{SSD}}). \tag{33}$$

If $C_{\text{SSD}} = 0$, then $C_{\text{T-SSD}} = 0$, and the proof is complete. Suppose instead that $C_{\text{SSD}} \neq 0$, with full column rank, cf. Theorem 2.2(a). We use induction to prove that $\mathcal{R}(C_{\text{SSD}}) \subseteq \mathcal{R}(C_i)$, where C_i is the internal matrix of the T-SSD algorithm for $i \in \{0, \dots, L\}$ and L is the iteration at which it terminates. When $i = 0$, the columns of $C_0 = I_{N_d}$ span $\mathbb{R}^{N_d}$ and, therefore, $\mathcal{R}(C_{\text{SSD}}) \subseteq \mathcal{R}(C_0)$. Assume then that $\mathcal{R}(C_{\text{SSD}}) \subseteq \mathcal{R}(C_i)$ for $i \in \{0, \dots, L-1\}$, and let us prove that $\mathcal{R}(C_{\text{SSD}}) \subseteq \mathcal{R}(C_{i+1})$.

Based on Theorem 2.2(b), we have $\mathcal{R}(D(X)C_{\text{SSD}}) = \mathcal{R}(D(Y)C_{\text{SSD}})$. This, together the definition of matrices A_0, B_0 in Algorithm 2 and the fact that $\mathcal{R}(C_{\text{SSD}}) \subseteq \mathcal{R}(C_i)$, yields

$$\mathcal{P}_{A_0 C_i} w = \mathcal{P}_{B_0 C_i} w = w, \tag{34}$$

for all $w \in \mathcal{R}(A_0 C_{\text{SSD}}) = \mathcal{R}(B_0 C_{\text{SSD}})$. Now, since $A_i = A_0 C_i$ and $B_i = B_0 C_i$ at iteration $i+1$ of the T-SSD algorithm, $G_{i+1} v = \mathcal{P}_{A_0 C_i} v - \mathcal{P}_{B_0 C_i} v$, for all $v \in \mathbb{R}^N$. This, together with (34), implies that $\mathcal{R}(A_0 C_{\text{SSD}}) = \mathcal{R}(B_0 C_{\text{SSD}}) \subseteq \text{null}(G_{i+1})$. Since $\epsilon = 0$, from Step 7 we know that V_{i+1} is a basis for $\text{null}(G_{i+1})$, and therefore

$$\mathcal{R}(A_0 C_{\text{SSD}}) = \mathcal{R}(B_0 C_{\text{SSD}}) \subseteq \mathcal{R}(V_{i+1}). \tag{35}$$

By the induction hypothesis $\mathcal{R}(C_{\text{SSD}}) \subseteq \mathcal{R}(C_i)$. This, together with the fact that C_{SSD} and C_i have full column rank (the latter because of Lemma 5.4(b)), implies that there exists a matrix F_i with full column rank such that

$$C_{\text{SSD}} = C_i F_i. \tag{36}$$

Using now (35)-(36) together with the fact that $A_i = A_0 C_i$, $B_i = B_0 C_i$, one can invoke Proposition 5.2(c) to deduce that $\mathcal{R}(F_i) \subseteq \mathcal{R}(E_{i+1})$. Consequently,

$$\mathcal{R}(C_{\text{SSD}}) = \mathcal{R}(C_i F_i) \subseteq \mathcal{R}(C_i E_{i+1}) = \mathcal{R}(C_{i+1}).$$

Hence, the induction is complete and

$$\mathcal{R}(C_{\text{SSD}}) \subseteq \mathcal{R}(C_i), \forall i \in \{1, \dots, L\}. \tag{37}$$

Since C_{SSD} is nonzero and has full column rank, one can deduce that C_L is nonzero and has full column rank as a result of Lemma 5.4(c). Consequently, the T-SSD algorithm terminates by executing Steps 16-17. Therefore, $C_{\text{T-SSD}} = C_L$ and using (33) and (37), we have $\mathcal{R}(C_{\text{T-SSD}}) = \mathcal{R}(C_{\text{SSD}})$ and consequently $\text{span}(D_{\text{T-SSD}}) = \text{span}(D_{\text{SSD}})$. The rest of the statement follows from Lemma 7.1. □

It is worth mentioning that, when implementing T-SSD for $\epsilon = 0$, we have found it useful to set ϵ to be a small positive number (instead of zero) to avoid complications by round-off errors.

Remark 7.4 *(Dynamical Properties of T-SSD Subspace with $\epsilon = 0$):* Given Theorem 7.3, the subspace identified by T-SSD for $\epsilon = 0$ enjoys important dynamical properties, cf. Section 2.4: under reasonable conditions on the density of data sampling, cf.[14, Theorems 5.7-5.8], the identified subspace is the maximal Koopman-invariant subspace in the span of the dictionary almost surely. Moreover, the eigenfunctions and predictors identified by T-SSD are almost surely exact. □

8 Efficient Implementation of T-SSD

Here, we propose a modification to the implementation of the T-SSD algorithm on digital computers to increase efficiency. This is based on the following observation: a close look at the form of the matrix $G_i \in \mathbb{R}^{N \times N}$ in Step 6 of Algorithm 2 as a difference of projections reveals that its eigenvectors are either in or orthogonal to the subspace $\mathcal{R}(A_{i-1}) + \mathcal{R}(B_{i-1})$, see e.g., [2]. However, in Step 8, the matrix E_i satisfies $\mathcal{R}(A_{i-1}E_i), \mathcal{R}(B_{i-1}E_i) \subseteq \mathcal{R}(V_i)$. Hence, the Symmetric-Intersection function filters out all eigenvectors of G_i that are orthogonal to $\mathcal{R}(A_{i-1}) + \mathcal{R}(B_{i-1})$, i.e., these eigenvectors are never used. This is despite the fact that, since generally $N \gg N_d$, such eigenvectors form a majority of eigenvectors of G_i (at least $N - 2N_d$ out of N).

This motivates us to seek a method that bypasses the calculation of the unused eigenvectors of G_i. To achieve this goal, let H_i be a matrix such that

$$\mathcal{R}(H_i) := \mathcal{R}([A_{i-1}, B_{i-1}]), \quad H_i^T H_i = I_{\sharp\text{cols}(H_i)}. \tag{38}$$

The columns of H_i form an orthonormal basis of $\mathcal{R}(A_{i-1}) + \mathcal{R}(B_{i-1})$. One can calculate H_i by applying the Gram–Schmidt process, or other closely related orthogonal factorization method such as QR decomposition (see e.g. [40]), on $[A_{i-1}, B_{i-1}]$. The next result shows that the eigendecomposition of the matrix $H_i^T G_i H_i$ completely captures the eigendecomposition of G_i on $\mathcal{R}(A_{i-1}) + \mathcal{R}(B_{i-1})$.

Proposition 8.1 *(Eigenvectors of $H_i^T G_i H_i$ Characterize All Eigenvectors of G_i in $\mathcal{R}(A_{i-1}) + \mathcal{R}(B_{i-1})$): Let G_i as defined in Step 6 of Algorithm 2, and let H_i satisfy* (38). *Then, $w \in \mathbb{C}^{N_d} \setminus \{0\}$ is an eigenvector of $H_i^T G_i H_i$ with eigenvalue λ if and only if $v = H_i w$ is an eigenvector of G_i with eigenvalue λ.*

PROOF. ($\Leftarrow$) By hypothesis, $G_i H_i w = \lambda H_i w$. Hence, $H_i^T G_i H_i w = \lambda H_i^T H_i w = \lambda w$ (where we have used (38)).

($\Rightarrow$) By hypothesis, $H_i^T G_i H_i w = \lambda w$. Using (38), this can be rewritten as

$$H_i^T (G_i H_i w - \lambda H_i w) = 0. \tag{39}$$

By definition of G_i, we can write $G_i H_i w = \mathcal{P}_{A_{i-1}}(H_i w) - \mathcal{P}_{B_{i-1}}(H_i w)$. From (38), we have $\mathcal{R}(A_{i-1}), \mathcal{R}(B_{i-1}) \subseteq \mathcal{R}(H_i)$. Since $\mathcal{P}_{A_{i-1}}(H_i w) \in \mathcal{R}(A_{i-1})$ and $\mathcal{P}_{B_{i-1}}(H_i w) \in \mathcal{R}(B_{i-1})$, we deduce that $G_i H_i w \in \mathcal{R}(H_i)$, and consequently, $G_i H_i w - \lambda H_i w \in \mathcal{R}(H_i)$. However, from (39), $(G_i H_i w - \lambda H_i w) \in \text{null}(H_i^T)$. Therefore, since $\mathcal{R}(H_i) \perp \text{null}(H_i^T)$, we conclude $G_i H_i w - \lambda H_i w = 0$, as claimed. □

Based on Proposition 8.1, we modify T-SSD to achieve higher computational efficiency. Formally, the **Efficient T-SSD** algorithm replaces Steps 6 and 8 of Algorithm 2 by

```
6.a:  $H_i \leftarrow \text{basis}([A_i, B_i])$
6.b:  $G_i \leftarrow H_i^T (A_{i-1}A_{i-1}^\dagger - B_{i-1}B_{i-1}^\dagger) H_i$
  8:  $E_i \leftarrow \text{Symmetric-Intersection}(H_i V_i, A_{i-1}, B_{i-1})$
```

These steps bypass the computation of the (unused) eigenvectors of G_i that are orthogonal to $\mathcal{R}(A_{i-1}) + \mathcal{R}(B_{i-1})$ in the original T-SSD implementation.

Remark 8.2 *(Computational Complexity of Efficient T-SSD):* Given N data snapshots and N_d dictionary functions, and considering the complexity of scalar operations as $O(1)$, the most time-consuming steps of Efficient T-SSD are calculating H_i in (38) and the null space calculations in the function Symmetric-Intersection, which can be done in $O(NN_d^2)$ operations. Since the algorithm terminates after at most N_d iterations, cf. Proposition 5.3, the overall complexity is $O(NN_d^3)$. Compared to T-SSD, cf. Remark 5.5, the efficient T-SSD algorithm provides a reduction of $O(N^2N_d^{-2})$, leading to a drastic reduction in run time for typical situations, where $N \gg N_d$. □

9 Simulation Results

Here, we illustrate the effectiveness of our proposed methods using three examples.

9.1 Hopf Normal Form

Consider the system [4, 29] on $\mathcal{M} = [-2, 2]^2$,

$$\begin{aligned} \dot{x_1} &= x_1 + 2x_2 - x_1(x_1^2 + x_2^2), \\ \dot{x_2} &= -2x_1 + x_2 - x_2(x_1^2 + x_2^2), \end{aligned} \tag{40}$$

with state $x = [x_1, x_2]^T$, which admits an attractive periodic orbit. We consider the discretized version of (40) with time step $\Delta t = 0.01s$ and gather $N = 10^4$ data snapshots in matrices X and Y, with initial conditions uniformly selected from $\mathcal{M}$. We consider the space of all polynomials up to degree 10 spanned by all the $N_d = 66$ distinct monomials in the form of $\prod_{i=1}^{10} \alpha_i$, with $\alpha_i \in \{1, x_1, x_2\}$ for $i \in \{1, \dots, 10\}$. To ensure resilience against machine precision errors and providing informative representations, we choose a dictionary D with $N_d = 66$ functions such that the columns of $D(X)$ are orthonormal [5].

We implement the Efficient T-SSD algorithm, cf. Section 8, with $\epsilon \in \{0.02, 0.05, 0.1, 0.15, 0.2\}$. Table 1 shows the dimension of the identified dictionary, $D_{\text{T-SSD}}$, versus the value of the design parameter ϵ. For $\epsilon = 0.2$, T-SSD identifies the original dictionary, certifying that the range spaces of $D(X)$ and $D(Y)$ are 0.2-apart. On the other hand, the one-dimensional subspace identified by $\epsilon = 0.02$ is in fact the maximal Koopman-invariant subspace of span(D), spanned by the trivial eigenfunction $\phi(x) \equiv 1$ with eigenvalue $\lambda = 1$.

Table 1
Dimension of subspace identified by Efficient T-SSD vs ϵ for (40).

ϵ	0.02	0.05	0.10	0.15	0.20
dim $D_{\text{T-SSD}}$	1	6	8	16	66

To demonstrate the effectiveness of the T-SSD algorithm in approximating Koopman eigenfunctions and invariant subspaces, we focus on the subspace identified with $\epsilon = 0.05$. In accordance with Proposition 6.6, T-SSD identifies the trivial eigenfunction $\phi(x) \equiv 1$ spanning the maximal Koopman-invariant subspace of span(D). T-SSD also approximates another real-valued eigenfunction with eigenvalue $\lambda = 0.9066$, whose absolute value is illustrated in Figure 2(right). Given that $|0.9066| < 1$, this eigenfunction predicts the existence of a forward invariant set (the periodic orbit in Figure 2(left)) at its zero-level set.

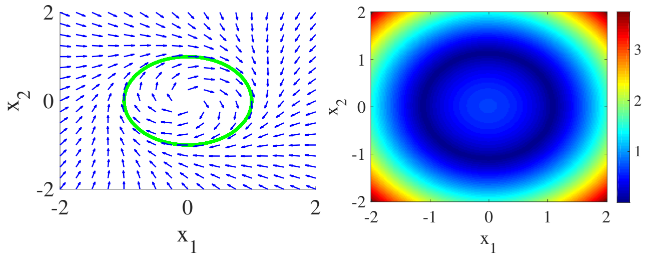

Figure 2. Vector field and limit cycle of system (40) (left) and the absolute value of eigenfunction with eigenvalue $\lambda = 0.9066$ (right).

In addition, T-SSD also identifies two pairs of complex eigenfunctions. For space reasons, we only show in Figure 3 one eigenfunction with eigenvalue $\lambda = 0.9938 + 0.0195j$ (the one closest to the unit circle). Its phase characterizes the oscillation of the trajectories in the state space.

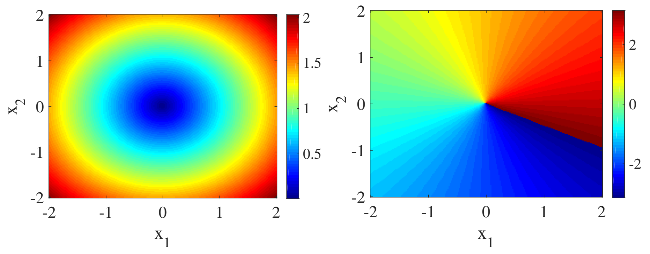

Figure 3. Absolute value (left) and phase (right) of the eigenfunction with eigenvalue $\lambda = 0.9938 + 0.0195j$ for (40).

To illustrate the efficacy of our algorithm regarding the prediction accuracy of the dictionary, we consider the relative linear prediction error associated with a dictionary $\mathcal{D}$ at point x given data snapshot matrices X and Y defined by

$$E_{\text{relative}}(x) := \frac{\|\mathcal{D} \circ T(x) - \mathcal{D}(x)K\|_2}{\|\mathcal{D} \circ T(x)\|_2} \times 100, \tag{41}$$

where $K = \text{EDMD}(\mathcal{D}, X, Y)$. Figure 4 compares this error on the state space $\mathcal{M}$ for the dictionary $D_{\text{T-SSD}}$ identified by T-SSD with $\epsilon = 0.05$ and for the original dictionary D. This error is evaluated at points other than the training data X and clearly shows the advantage of $D_{\text{T-SSD}}$ over the original dictionary both in prediction errors and capturing the radial symmetry of the vector field.

Noting that the error in (41) depends on the dictionary and does not provide information about the subspace it spans (or the individual members of the subspace), we also consider the latter. For this reason, we use the data sampling strategy used earlier to build a test data set denoted by snapshot matrices X_{test} and Y_{test} with $N_{\text{test}} = 10^4$ samples. Given a dictionary $\mathcal{D}$, we evaluate the invariance proximity of span$(\mathcal{D})$ as the smallest ϵ such that $\mathcal{R}(\mathcal{D}(X_{\text{test}}))$ and $\mathcal{R}(\mathcal{D}(Y_{\text{test}}))$ are ϵ-apart. This data-driven measure is equivalent to the maximum relative root mean square error for a function in the

[5] This dictionary can be found by first forming a dictionary comprised of the monomials and then performing a linear transformation on the dictionary to make the columns orthonormal. The linear transformation does not impact the captured dynamical information (cf. Lemma 7.1).

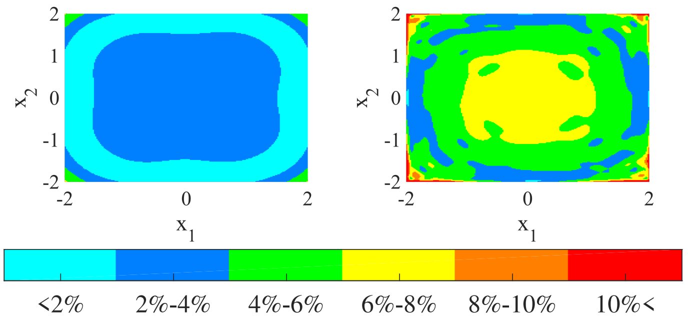

Figure 4. Relative linear prediction error for dictionary identified by T-SSD ($\epsilon = 0.05$) (left) and the original dictionary (right for (40).

span of a given dictionary $\mathcal{D}$ on the test data defined as

$$\mathrm{RRMSE}_{\max}(\mathcal{D}, X_{\text{test}}, Y_{\text{test}}) = \max_{f \in \operatorname{span}(\mathcal{D})} \frac{\sqrt{\frac{1}{N_t}\sum_{i=1}^{N_t} |\mathcal{K}f(x_i) - \mathfrak{P}^{\text{T-SSD}}_{\mathcal{K}f}|^2}}{\sqrt{\frac{1}{N_t}\sum_{i=1}^{N_t} |\mathcal{K}f(x_i)|^2}}, \tag{42}$$

where x_i^T and y_i^T correspond to the ith rows of X_{test} and Y_{test} respectively, $y_i = T(x_i)$, and T is the map defining the dynamics (40). The predictor $\mathfrak{P}^{\text{T-SSD}}_{\mathcal{K}f}$ is defined in (23) and calculated based on the test data. It is important to note that the evaluation of the error in (42) goes beyond the assumptions of Theorem 6.2, since the dictionary is identified with the original data X, Y, but the error is evaluated on the test data $X_{\text{test}}, Y_{\text{test}}$ (instead, the guarantee of Theorem 6.2 are only valid when the error is evaluated on the original data). Following the reasoning in the proof of Theorem 6.2 and Definition 4.1, one can analytically show that

$$\mathrm{RRMSE}_{\max}(\mathcal{D}, X_{\text{test}}, Y_{\text{test}}) = \lambda_{\max}(\mathcal{P}_{\mathcal{D}(X_{\text{test}})} - \mathcal{P}_{\mathcal{D}(Y_{\text{test}})}),$$

where $\lambda_{\max}$ denotes the largest eigenvalue of the argument. Table 2 shows the maximum relative root mean square error for the subspaces identified by T-SSD given different values of ϵ. According to Table 2, despite the fact that we have used different data for identification and evaluation, the error on the test data satisfies the upper bound accuracy requirement enforced by the accuracy parameter ϵ .

Table 2
Maximum Relative Root Mean Square Error vs ϵ for (40).

$\boldsymbol{\epsilon}$	0.02	0.05	0.10	0.15	0.20
$\mathrm{RRMSE}_{\max}$	~ 0	0.037	0.100	0.115	0.185

9.2 Duffing System

Consider the Duffing system [42] on $\mathcal{M} = [-2, 2]^2$,

$$\begin{aligned} \dot{x_1} &= x_2, \\ \dot{x_2} &= -0.5x_2 + x_1(1 - x_1^2), \end{aligned} \tag{43}$$

with state $x = [x_1, x_2]^T$, which has an unstable equilibrium at the origin and two asymptotically stable equilibria at $(-1, 0)$ and $(1, 0)$. We consider the discretized version of (43) with time step $\Delta t = 0.02s$ and gather $N = 10^4$ data snapshots in matrices X and Y from 5000 trajectories with length equal to two time steps and initial conditions uniformly selected from $\mathcal{M}$. Similarly to the previous example, we use a dictionary D with $N_d = 66$ elements spanning the space of all polynomials up to degree 10 such that the columns of $D(X)$ are orthonormal.

We apply the Efficient T-SSD algorithm, cf. Section 8, with $\epsilon \in \{0.01, 0.02, 0.08, 0.14, 0.2, 0.26\}$. Table 3 shows the dimension of the identified dictionary, $D_{\text{T-SSD}}$, versus the value of the design parameter ϵ. For $\epsilon = 0.26$, T-SSD identifies the original dictionary, certifying that the range spaces of $D(X)$ and $D(Y)$ are 0.26-apart. On the other hand, the one-dimensional subspace identified by $\epsilon = 0.01$ is in fact the maximal Koopman-invariant subspace of span(D), spanned by the trivial eigenfunction $\phi(x) \equiv 1$ with eigenvalue $\lambda = 1$.

Table 3
Dimension of subspace identified by Efficient T-SSD vs ϵ for (43).

$\boldsymbol{\epsilon}$	0.01	0.02	0.08	0.14	0.20	0.26
dim $\boldsymbol{D_{\text{T-SSD}}}$	1	2	20	44	58	66

To demonstrate the effectiveness of the T-SSD algorithm in approximating Koopman eigenfunctions and invariant subspaces, we focus on the subspace identified with $\epsilon = 0.02$. Consistent with Proposition 6.6, T-SSD identifies the trivial eigenfunction $\phi(x) \equiv 1$ spanning the maximal Koopman-invariant subspace of span(D). T-SSD also approximates another real-valued eigenfunction with eigenvalue $\lambda = 0.9839$ depicted in Figure 5(right), which clearly captures the attractiveness of the asymptotically stable equilibria and the general behavior of the vector field depicted in Figure 5(left).

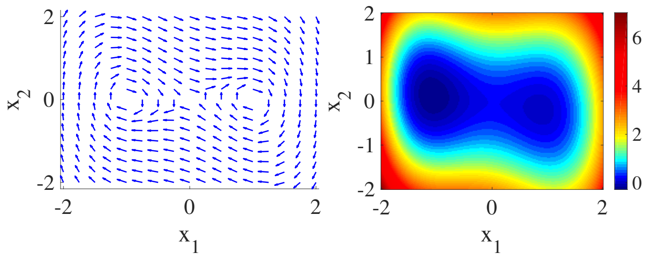

Figure 5. Vector field (left) and eigenfunction with eigenvalue $\lambda = 0.9839$ (right) for (43).

To illustrate the efficacy of our algorithm regarding the prediction accuracy of the dictionary, Figure 6 compares the relative linear prediction error (41) on the state space $\mathcal{M}$ for the dictionary $D_{\text{T-SSD}}$ identified by T-SSD with $\epsilon = 0.02$ and for the original dictionary D evaluated at out-of-sample points other than X. Figure 6 clearly shows the effectiveness of the T-SSD algorithm in improving the prediction accuracy.

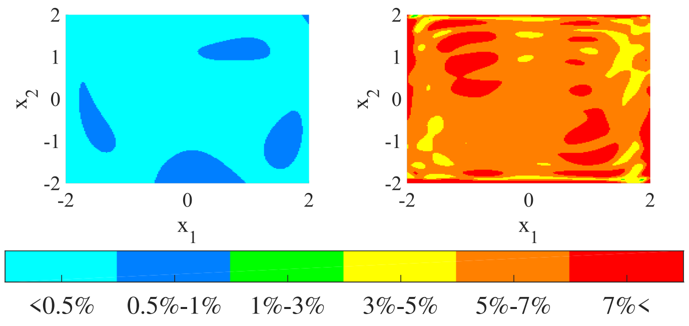

Figure 6. Relative linear prediction error for dictionary identified by T-SSD ($\epsilon = 0.02$) (left) and the original dictionary (right) for (43).

To analyze the error for individual functions in the identified subspaces by T-SSD, we form a random test data set X_{test}, Y_{test} gathered with the same number of elements and sampling strategy used for X and Y. Table 4 shows the maximum relative root mean square error defined in (42) for the subspaces identified by T-SSD given different values of ϵ. Despite the fact that we use different data for identification and evaluation, Table 4 shows the effectiveness of the T-SSD algorithm for identifying subspaces on which all functions have prediction errors characterized by the accuracy parameter ϵ.

Table 4
Maximum Relative Root Mean Square Error vs ϵ for (43).

$\boldsymbol{\epsilon}$	0.01	0.02	0.08	0.14	0.20	0.26
$\text{RRMSE}_{\max}$	~ 0	0.004	0.054	0.123	0.190	0.236

9.3 Consensus on Harmonic Mean

Given N_a agents with state $x = [x_1, \ldots, x_{N_a}]^T$ communicating through a graph with adjacency matrix A, consider the dynamics

$$\dot{x}_i = N_a\, x_i^2\, \Upsilon(x)^{-2} \sum_{j=1}^{N_a} a_{ij}(x_j - x_i),\ i \in \{1, \ldots, N_a\}, \quad (44)$$

where a_{ij} is the element of A on row i and column j and $\Upsilon(x)$ is the harmonic mean of the state elements defined as

$$\Upsilon(x) = N_a \Big(\sum_{k=1}^{N_a} x_k^{-1} \Big)^{-1}.$$

For any initial condition x_0, all the state elements converge to the harmonic mean of the initial condition $\Upsilon(x_0)$ [7, Proposition 10], i.e., the agents achieve consensus on $\Upsilon(x_0)$. For the purpose of this example, we consider $N_a = 5$ agents communicating through an undirected ring graph and states belonging to the state space $\mathcal{M} = [1, 5]^5$. We consider the discretized version of (44) with time step $\Delta t = 0.01s$ and gather $N = 4 \times 10^4$ data snapshots in matrices X and Y from 2×10^4 trajectories with length equal to two time steps and initial conditions uniformly selected from $\mathcal{M}$. For our dictionary, we consider the space of all polynomials up to degree 6 and choose a dictionary D with $N_d = 462$ functions spanning the space such that the columns of $D(X)$ are orthonormal.

We apply the Efficient T-SSD algorithm, cf. Section 8, with $\epsilon \in \{0.05, 0.15, 0.3, 0.55, 0.8\}$. Table 5 shows the dimension of the identified dictionary, $D_{\text{T-SSD}}$, versus the value of the design parameter ϵ. For $\epsilon = 0.8$, T-SSD identifies the original subspace, certifying that the range spaces of $D(X)$ and $D(Y)$ are 0.8-apart. On the other hand, the one-dimensional subspace identified by $\epsilon = 0.05$ is in fact the maximal Koopman-invariant subspace of span(D), spanned by the trivial eigenfunction $\phi(x) \equiv 1$ with eigenvalue $\lambda = 1$.

Table 5
Dimension of subspace identified by Efficient T-SSD vs ϵ for (44).

$\boldsymbol{\epsilon}$	0.05	0.15	0.30	0.55	0.80
dim $\boldsymbol{D}_{\textbf{T-SSD}}$	1	14	64	272	462

To illustrate the efficacy of T-SSD algorithm regarding prediction accuracy, we first form a test data set comprised of snapshots matrices X_{test} and Y_{test} sampled with the same sampling strategy and number of samples as X and Y. Figure 7 provides histogram plots comparing the relative prediction error defined in (41) of the dictionary identified by T-SSD with $\epsilon = 0.15$ and the original dictionary applied on the test data. In Figure 7 the horizontal axis denotes the prediction error while the vertical axis shows the percentage of test data per interval. Figure 7 clearly shows the effectiveness of the T-SSD algorithm in improving the prediction accuracy.

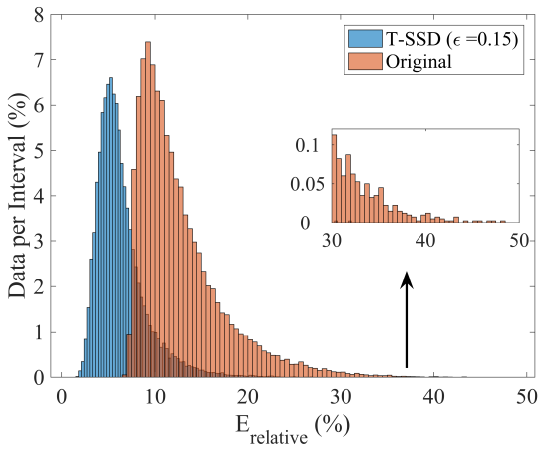

Figure 7. Relative linear prediction error on test data for the dictionary identified with T-SSD ($\epsilon = 0.15$) and the original dictionary.

We also consider the prediction accuracy for the individual functions. Table 6 shows the maximum relative root mean square error defined in (42) for the subspaces identified by T-SSD given different values of ϵ. According to Table 6, despite using different data for identification and evaluation, the error on the test data satisfies the upper bound accuracy requirement enforced by the accuracy parameter ϵ.

Table 6
Maximum Relative Root Mean Square Error vs ϵ for (44).

$\boldsymbol{\epsilon}$	0.05	0.15	0.30	0.55	0.8
$\text{RRMSE}_{\max}$	~ 0	0.144	0.295	0.549	0.769

10 Conclusions

We have presented the T-SSD algorithm, a data-driven strategy that employs data snapshots from an unknown dynamical system to refine a given dictionary of functions, yielding a subspace close to being invariant under the Koopman operator. A design parameter allows to balance the prediction accuracy and expressiveness of the algorithms' output, which always contains the maximal Koopman-invariant subspace and all Koopman eigenfunctions in the span of the original dictionary. The proposed algorithm generalizes both Extended Dynamic Mode Decomposition and Symmetric Subspace Decomposition. Future work will investigate noise-resilient strategies to approximate Koopman-invariant subspaces and methods to construct expressive dictionaries with high accuracy by alternating between growing the set of functions (using specific basic functions or neural networks) and pruning the dictionary to enhance accuracy while providing accuracy bounds for all members of the identified vector space of functions. Moreover, we aim to explore the application of the proposed method in stability analysis, data-driven construction of Lyapunov functions [6], and designing control schemes with formal performance and stability guarantees by using the Koopman operator to model control systems as bilinear or switched-linear systems.

References

[1] P. A. Absil, R. Mahony, and R. Sepulchre. *Optimization algorithms on matrix manifolds*. Princeton University Press, 2009.

[2] W. N. Anderson Jr, E. J. Harner, and G. E. Trapp. Eigenvalues of the difference and product of projections. *Linear and Multilinear Algebra*, 17(3-4):295–299, 1985.

[3] S. L. Brunton, B. W. Brunton, J. L. Proctor, and J. N. Kutz. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PLOS One*, 11(2):1–19, 2016.

[4] 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*, 113(15):3932–3937, 2016.

[5] M. Budišić, R. Mohr, and I. Mezić. Applied Koopmanism. *Chaos*, 22(4):047510, 2012.

[6] H. Choi, U. Vaidya, and Y. Chen. A convex data-driven approach for nonlinear control synthesis. *arXiv preprint arXiv:2006.15477*, 2020.

[7] J. Cortés. Distributed algorithms for reaching consensus on general functions. *Automatica*, 44(3):726–737, 2008.

[8] S. T. M. Dawson, M. S. Hemati, M. O. Williams, and C. W. Rowley. Characterizing and correcting for the effect of sensor noise in the dynamic mode decomposition. *Experiments in Fluids*, 57(3):42, 2016.

[9] C. Folkestad, Y. Chen, A. D. Ames, and J. W. Burdick. Data-driven safety-critical control: Synthesizing control barrier functions with Koopman operators. *IEEE Control Systems Letters*, 5(6):2012–2017, 2020.

[10] D. Goswami and D. A. Paley. Bilinearization, reachability, and optimal control of control-affine nonlinear systems: A Koopman spectral approach. *IEEE Transactions on Automatic Control*, 2021. To appear.

[11] M. Haseli and J. Cortés. Approximating the Koopman operator using noisy data: noise-resilient extended dynamic mode decomposition. In *American Control Conference*, pages 5499–5504, Philadelphia, PA, July 2019.

[12] M. Haseli and J. Cortés. Data-driven approximation of Koopman-invariant subspaces with tunable accuracy. In *American Control Conference*, pages 469–474, New Orleans, LA, July 2021.

[13] M. Haseli and J. Cortés. Parallel learning of Koopman eigenfunctions and invariant subspaces for accurate long-term prediction. *IEEE Transactions on Control of Network Systems*, 8(4):1833–1845, 2021.

[14] M. Haseli and J. Cortés. Learning Koopman eigenfunctions and invariant subspaces from data: Symmetric Subspace Decomposition. *IEEE Transactions on Automatic Control*, 67 (7):3442–3457, 2022.

[15] C. A. Johnson and E. Yeung. A class of logistic functions for approximating state-inclusive Koopman operators. In *American Control Conference*, pages 4803–4810, Milwaukee, WI, June 2018. IEEE.

[16] E. Kaiser, J. N. Kutz, and S. L. Brunton. Data-driven discovery of Koopman eigenfunctions for control. *Machine Learning: Science and Technology*, 2(3):035023, 2021.

[17] S. Klus, P. Koltai, and C. Schütte. On the numerical approximation of the Perron-Frobenius and Koopman operator. *Journal of Computational Dynamics*, 3(1):51–79, 2016.

[6] The recent work [28, Section V.C] provides an interesting example of using the T-SSD algorithm as a subroutine precisely for this purpose.

[18] S. Klus, F. Nüske, S. Peitz, J. H. Niemann, C. Clementi, and C. Schütte. Data-driven approximation of the Koopman generator: Model reduction, system identification, and control. *Physica D: Nonlinear Phenomena*, 406:132416, 2020.

[19] B. O. Koopman. Hamiltonian systems and transformation in Hilbert space. *Proceedings of the National Academy of Sciences*, 17(5):315–318, 1931.

[20] B. O. Koopman and J. V. Neumann. Dynamical systems of continuous spectra. *Proceedings of the National Academy of Sciences*, 18(3):255–263, 1932.

[21] M. Korda and I. Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93:149–160, 2018.

[22] M. Korda and I. Mezić. On convergence of extended dynamic mode decomposition to the Koopman operator. *Journal of Nonlinear Science*, 28(2):687–710, 2018.

[23] M. Korda and I. Mezic. Optimal construction of Koopman eigenfunctions for prediction and control. *IEEE Transactions on Automatic Control*, 65(12):5114–5129, 2020.

[24] H. Lu and D. M. Tartakovsky. Prediction accuracy of dynamic mode decomposition. *SIAM Journal on Scientific Computing*, 42(3):A1639–A1662, 2020.

[25] B. Lusch, J. N. Kutz, and S. L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9(1):1–10, 2018.

[26] G. Mamakoukas, M. Castano, X. Tan, and T. Murphey. Local Koopman operators for data-driven control of robotic systems. In *Robotics: Science and Systems*, Freiburg, Germany, June 2019.

[27] G. Mamakoukas, M. L. Castano, X. Tan, and T. D. Murphey. Derivative-based Koopman operators for real-time control of robotic systems. *IEEE Transactions on Robotics*, 2021.

[28] G. Mamakoukas, I. Abraham, and T. D. Murphey. Learning stable models for prediction and control. *https://arxiv.org/abs/2005.04291v2*, 2022.

[29] J. E. Marsden and M. McCracken. *The Hopf bifurcation and its applications*, volume 19. Springer Science & Business Media, 2012.

[30] A. Mauroy and I. Mezić. Global stability analysis using the eigenfunctions of the Koopman operator. *IEEE Transactions on Automatic Control*, 61(11):3356–3369, 2016.

[31] I. Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1-3): 309–325, 2005.

[32] F. Nüske, S. Peitz, F. Philipp, M. Schaller, and K. Worthmann. Finite-data error bounds for Koopman-based prediction and control. *arXiv preprint arXiv:2108.07102*, 2021.

[33] S. E. Otto and C. W. Rowley. Linearly recurrent autoencoder networks for learning dynamics. *SIAM Journal on Applied Dynamical Systems*, 18(1):558–593, 2019.

[34] S. Pan, N. Arnold-Medabalimi, and K. Duraisamy. Sparsity-promoting algorithms for the discovery of informative Koopman-invariant subspaces. *Journal of Fluid Mechanics*, 917, 2021.

[35] S. Peitz and S. Klus. Koopman operator-based model reduction for switched-system control of PDEs. *Automatica*, 106: 184–191, 2019.

[36] C. W. Rowley, I. Mezić, S. Bagheri, P. Schlatter, and D. S. Henningson. Spectral analysis of nonlinear flows. *Journal of Fluid Mechanics*, 641:115–127, 2009.

[37] P. J. Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010.

[38] S. H. Son, A. Narasingam, and J. S. Kwon. Handling plant-model mismatch in Koopman Lyapunov-based model predictive control via offset-free control framework. *arXiv preprint arXiv:2010.07239*, 2020.

[39] N. Takeishi, Y. Kawahara, and T. Yairi. Learning Koopman invariant subspaces for dynamic mode decomposition. In *Conference on Neural Information Processing Systems*, pages 1130–1140, 2017.

[40] L. N. Trefethen and D. Bau. *Numerical Linear Algebra.* SIAM, Philadelphia, PA, 1997.

[41] J. H. Tu, C. W. Rowley, D. M. Luchtenburg, S. L. Brunton, and J. N. Kutz. On dynamic mode decomposition: theory and applications. *Journal of Computational Dynamics*, 1(2): 391–421, 2014.

[42] M. O. Williams, I. G. Kevrekidis, and C. W. Rowley. A data-driven approximation of the Koopman operator: Extending dynamic mode decomposition. *Journal of Nonlinear Science*, 25(6):1307–1346, 2015.

[43] E. Yeung, S. Kundu, and N. Hodas. Learning deep neural network representations for Koopman operators of nonlinear dynamical systems. In *American Control Conference*, pages 4832–4839, Philadelphia, PA, July 2019.

[44] C. Zhang and E. Zuazua. A quantitative analysis of Koopman operator methods for system identification and predictions. *https://hal.archives-ouvertes.fr/hal-03278445*, 2021.

[45] V. Zinage and E. Bakolas. Koopman operator based modeling for quadrotor control on SE(3). *IEEE Control Systems Letters*, 6:752–757, 2022.

A Basic Algebraic Results

Here we collect two algebraic results from [14] that are used in our technical treatment.

Lemma A.1 *([14, Lemma A.1]): Let* $A, B \in \mathbb{R}^{m\times n}$ *be matrices with full column rank. Suppose that the columns of* $Z = [(Z^A)^T, (Z^B)^T]^T \in \mathbb{R}^{2n\times l}$ *form a basis for the null space of* $[A, B]$*, where* $Z^A, Z^B \in \mathbb{R}^{n\times l}$*. Then,*

(a) $\mathcal{R}(AZ^A) = \mathcal{R}(A) \cap \mathcal{R}(B)$*;*

(b) Z^A *and* Z^B *have full column rank.*

Lemma A.2 *([14, Lemma A.2]): Let* A, C, D *be matrices of appropriate sizes, with* A *having full column rank. Then* $\mathcal{R}(AC) \subseteq \mathcal{R}(AD)$ *if and only if* $\mathcal{R}(C) \subseteq \mathcal{R}(D)$*.*