POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

On latent dynamics learning in nonlinear reduced order modeling

Tesi di Laurea Magistrale in
Mathematical Engineering - Ingegneria Matematica

Nicola Farenga, 10537686

Abstract: In this work, we present the novel mathematical framework of *latent dynamics models* (LDMs) for reduced order modeling of parameterized nonlinear time-dependent PDEs. Our framework casts this latter task as a nonlinear dimensionality reduction problem, while constraining the latent state to evolve accordingly to an (unknown) dynamical system, namely a latent vector ordinary differential equation (ODE). A time-continuous setting is employed to derive error and stability estimates for the LDM approximation of the full order model (FOM) solution. We analyze the impact of using an explicit Runge-Kutta scheme in the time-discrete setting, resulting in the ΔLDM formulation, and further explore the learnable setting, ΔLDM_θ, where deep neural networks approximate the discrete LDM components, while providing a bounded approximation error with respect to the FOM. Moreover, we extend the concept of parameterized Neural ODE – recently proposed as a possible way to build data-driven dynamical systems with varying input parameters – to be a convolutional architecture, where the input parameters information is injected by means of an affine modulation mechanism, while designing a convolutional autoencoder neural network able to retain spatial-coherence, thus enhancing interpretability at the latent level. Numerical experiments, including the Burgers' and the advection-reaction-diffusion equations, demonstrate the framework's ability to obtain, in a multi-query context, a *time-continuous* approximation of the FOM solution, thus being able to query the LDM approximation at any given time instance while retaining a prescribed level of accuracy. Our findings highlight the remarkable potential of the proposed LDMs, representing a mathematically rigorous framework to enhance the accuracy and approximation capabilities of reduced order modeling for time-dependent parameterized PDEs.

Advisor:
Prof. Andrea Manzoni

Co-advisors:
Dr. Stefania Fresca
Dr. Simone Brivio

Academic year:
2023-2024

Key-words: Reduced order modeling, latent dynamics, deep learning

1 Introduction

The numerical solution of parameterized nonlinear time-dependent partial differential equations (PDEs) by means of traditional high-fidelity techniques [88, 96], e.g., the finite element method, entails large computational costs, primarily stemming from the large number of degrees of freedom (DoFs) involved in their spatial discretization, and further compounded by the need to select a temporal discretization that ensures the required level of accuracy [22]. As a result, the computationally intensive nature of such techniques prevents them to be employed in *real-time* and *multi-query* scenarios, where the PDE solution has to be computed for multiple parameters instances, within reasonable time-frames.
Reduced order models (ROMs) address such issue by introducing a low-dimensional representation approximating the solution of the high-fidelity, full order model (FOM), thereby mitigating the computational burden arising from the solution of the original high-dimensional problem. The core assumption underlying such techniques is that FOM solutions lie on a low-dimensional manifold, i.e., *the solution manifold*. In particular, two main tasks have to be performed in order to construct a ROM: *(i)* the computation of a suitable *trial manifold* approximating the solution manifold, namely the set of PDE's solutions by varying time and parameters, and *(ii)* the identification of a *latent dynamics* embedding the reduced representation onto the trial manifold. Traditional reduced basis (RB) methods [6, 18, 86] approximate the solution manifold by means of a *linear trial subspace*, spanned by a set of basis functions. In the parameterized setting, RB methods face significant challenges with *nonaffine* parameter dependencies, namely, when the operators arising from the problem's spatial discretization cannot be expressed as a linear combination of parameter-independent matrices weighted by parameter-dependent coefficients [78]. This nonaffine structure is present in many scenarios, for instance when the parameter dependence is nonlinear, in problems featuring geometrical deformations, as well as in multi-physics systems. In such settings, the components involved in the decomposition of the FOM operators needs to be re-assembled for each specific time and parameters instances, thus breaking the efficiency of the *online-offline* procedure. Moreover, RB methods rely on linear projections which limit their expressivity for problems exhibiting slowly-decaying *Kolmogorov n-width* [44, 62, 80, 83]. This implies that achieving substantial dimensionality reduction becomes impossible, as the intrinsic low-dimensional structure cannot be accurately captured via small linear subspaces. Furthermore, the intrusive nature of such techniques requires access to the high-fidelity operators for the construction of the *reduced order dynamics* governing the evolution of the reduced state. This aspect represents a restrictive requirement in those cases in which neither the algebraic-structure of the original problem nor the operators are accessible, or when black-box FOM solvers are employed.
The previous limitations have motivated the recent adoption of data-driven methods [51], addressing the common issues related to traditional linear ROMs via machine learning-based techniques. In the specific context of reduced order modeling for nonlinear parameterized time-dependent problems, multiple deep neural network (DNN)-based approaches have been proposed to address two primary tasks: *(i)* nonlinear dimensionality reduction, by means of convolutional autoencoders (AEs) [43, 69, 93] or geometry-informed techniques [31, 32]; and *(ii)* modeling the latent dynamics of the resulting time-evolving parameterized reduced representation, via regressive approaches [33, 35, 77, 82], recurrent strategies [21, 57], or by coupling these two modeling paradigms [34]. In particular, recent DL-based *latent dynamics learning* approaches have predominantly relied on recurrent neural networks (RNNs) -based architectures [99, 100] and autoregressive schemes [72, 109], aiming at modeling the temporal dependency of the latent state in a sequential manner.
However, both traditional and DL-based ROMs are intrinsically tied to the temporal discretization employed during the offline training phase. This aspect represents a critical constraint, restricting ROMs' ability to adapt to different temporal resolutions during the online phase. As a consequence, it is often necessary to rebuild the reduced basis or undergo additional fine-tuning stages, thereby entailing increased offline computational costs. To this end, a central aspect in reduced order modeling resides in whether the ROM solution is a meaningful *time-continuous approximation* of the FOM, in which case it could be possible to query the learned ROM at any given time by retaining a prescribed level of accuracy.
A recently proposed DL-based reduced order modeling approach consists in leveraging neural ordinary differential equations (NODEs) [16] coupled with AEs, resulting in the AE-NODE architecture, in order to implicitly learn the vector field parameterizing the latent dynamics [25, 70]. In particular, as high-

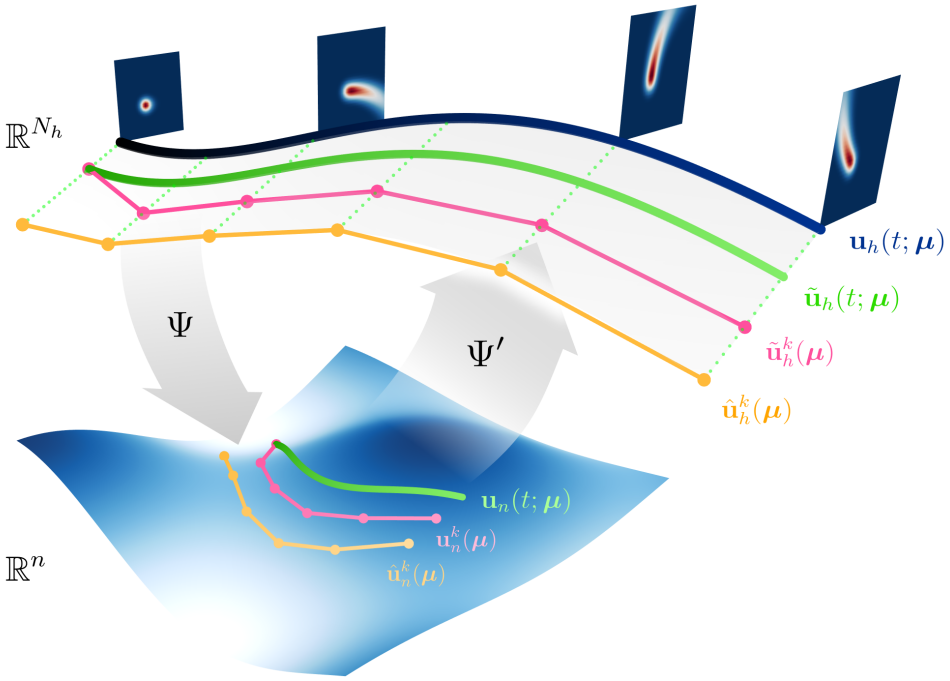

Figure 1: Different levels of approximation in the proposed LDM framework, illustrating: *(i)* the time-continuous full-order model $\mathbf{u}_h(t;\boldsymbol{\mu})$, the time-continuous LDM approximation $\tilde{\mathbf{u}}_h(t;\boldsymbol{\mu})$ with the associated latent dynamics $\mathbf{u}_n(t;\boldsymbol{\mu})$; *(ii)* the time-discrete approximation $\tilde{\mathbf{u}}_h^k(\boldsymbol{\mu})$ arising from the numerical solution of the latent dynamics $\mathbf{u}_n^k(\boldsymbol{\mu})$, leading to the notion of ΔLDM; *(iii)* the learned approximation $\hat{\mathbf{u}}_h^k(\boldsymbol{\mu})$ and the corresponding learned latent dynamics $\hat{\mathbf{u}}_n^k(\boldsymbol{\mu})$, associated to the learnable ΔLDM_θ. Here, Ψ and Ψ' denote the nonlinear projection and reconstruction maps, respectively.

lighted by [66], the *continuous-time*[1] nature of NODEs, employed in time-series modeling setting for the approximation of dynamical systems' evolution, have demonstrated promising capabilities at the task of zero-shot generalization across temporal discretizations, when adopting higher-order Runge-Kutta (RK) schemes (ODE-Nets). However, as already pointed out by [89], models characterized by a continuous-time inductive bias, such as ODE-Nets, do not necessarily guarantee that the learned representation will exhibit a time-continuous approximation property. This is usually due to the fact that learning settings are inherently time-discrete, thus possibly leading to overfitting with respect to the training temporal discretization. Despite the centrality of this aspect, the time-continuous approximation properties of the AE-NODE paradigm have not yet been addressed in a multi-query reduced order modeling context.

1.1 Contributions

In this work, we aim at:

- introducing the *latent dynamics model* (LDM) mathematical framework and rigorously demonstrating that it is able to provide an accurate and time-continuous approximation of the FOM solution, arising from the semi-discretization of parameterized nonlinear time-dependent PDEs;
- characterizing the resulting reduced order modeling framework based on the AE-NODE architecture [17, 25, 70, 71], which combines *(a)* autoencoders (AEs) to reduce the dimensionality of the state to a handful of latent coordinates, and *(b)* ODE-Nets to describe their dynamics.

Specifically, the problem of nonlinear dimensionality reduction for parameterized dynamical systems is first addressed, highlighting the role of the nonlinear projection and reconstruction maps and motivating

[1] For a detailed description of continuous-depth and continuous-time models, we refer to Appendix A.

the introduction of a unified framework addressing both the problem of dimensionality reduction and latent dynamics modeling. Thus, the novel concept of *latent dynamics model* (LDM) is introduced, as a mathematical framework formalizing the problem of computing an accurate and time-continuous approximation of the FOM solution which, at the best of our knowledge, is currently missing in the scientific literature.
The LDM framework is introduced in a time-continuous setting in order to consecutively analyze *(i)* the impact of employing a numerical integration time scheme for the solution of the latent dynamics, leading to a time-discrete LDM formulation, namely ΔLDM; *(ii)* the approximation capabilities in a learnable setting, where DNNs are adopted to approximate the components involved in the ΔLDM, via a ΔLDM_θ of learnable weights θ.
More precisely, at the time-continuous level, suitable error and stability estimates for the LDM approximation of the FOM solution are derived. In the time-discrete setting, the role of an explicit Runge-Kutta (RK) scheme in approximating the solution of the latent dynamics within the ΔLDM scheme is analyzed. In particular, convergence, consistency, and zero-stability results are provided, highlighting the inheritance of these properties by the high-dimensional ΔLDM approximation under a Lipschitz requirement on the nonlinear projection and reconstruction maps. Additionally, an error decomposition formula that accounts for the different error sources is derived. Finally, in the ΔLDM_θ learnable setting, the issue of approximating the discrete LDM components using DNNs within a parameterized AE-NODE architecture is addressed. Specifically, the approximation capabilities of the ΔLDM_θ with respect to the FOM solution are demonstrated through the derivation of a suitable error bound.
From a DL perspective, the novel architectural choices, underlying the proposed ΔLDM_θ formulation, are described, departing from the traditional AE-NODE setting. In particular, a fully-convolutional AE architecture, based on a prior interpolation operation, is considered. It aims at avoiding input reshape operations and exploiting spatial correlations, with the possibility to retain *spatially-coherent* information at the latent level. A parameterized convolutional NODE architecture is introduced for the approximation of the latent dynamics. Specifically, an affine-modulation scheme is adopted to include the parametric information into the latent representation.
The introduced LDM framework is tested on high-dimensional dynamical systems arising from the semi-discretization of parameterized nonlinear time-dependent PDEs. Specifically, the nonlinearity can either result from the presence of nonlinear terms into the governing equations or from nonlinear parametric dependencies, leading to nonaffine scenarios that are difficult to address using traditional ROMs. In particular, two problems are considered: *(i)* one-dimensional Burgers' equation, *(ii)* two-dimensional advection-reaction diffusion equation. These test cases are intended to demonstrate that the introduced framework exhibits a time-continuous approximation property, resulting in the capability of zero-shot generalization to finer temporal discretizations than the one employed at training time.
In summary, the paper is organized as follows: after an extensive literature review, Section 2 details the problem formulation and introduces the time-continuous approximation property. Section 3 presents the LDM framework in a time-continuous setting, and then covers the discrete and learnable LDM formulations. Section 4 discusses the architectural choices in the DL context. Numerical experiments and results are presented in Section 5. The paper concludes with Section 6, summarizing the main findings and outlining possible future directions.

1.2 Literature review

The concept of latent dynamics has been widely explored in the field of deep learning in order to properly model the evolution of a reduced representation (latent state) of high-dimensional temporally-evolving observations, with applications ranging from planning and control [48, 106] to computer graphics [38]. The concept emerged with the application of autoencoders (AEs) in a dynamical context, to reduce the dimensionality of time-evolving data, leading to the development of *dynamical* AEs [42]. In particular, modeling the temporal and parametric relations of reduced representations is a central aspect in the construction of ROMs, as highlighted in Section 2. In the following, we examine the main approaches used to model latent state temporal and parametric relations in the context of DL-based reduced-order and surrogate PDEs modeling, with a focus on parameterized multi-query settings.

Direct (regressive) approaches. A significant class of data-driven approaches models the latent dynamics as a map from the time-parameter space to the reduced manifold using regression-based frame-

works. These methods aim to directly learn such mappings via feedforward NNs [33, 103], explicitly capturing the dependence of the latent state on temporal and parametric inputs. This approach was extended in [35] by adopting a hybrid POD-AE architecture to reduce the offline computational costs associated with expensive training procedures, resulting in the POD-DL-ROM architecture. Additionally, the approximation capabilities of DL-ROMs have been analyzed in [10, 30]. Similar approaches are employed into the field of *operator learning* (OL) [65], where recent observations suggest that learning operators in the latent space leads to improvements in accuracy and computational efficiency [63]. In [11] the concepts of DL-ROMs and OL are combined within a physics-informed framework, where a fine-tuning strategy is proposed to improve generalization capabilities in a small data regime.

Recurrent & autoregressive approaches. The landscape of data-driven reduced order modeling methods based on latent dynamics learning has been primarily shaped by the adoption of recurrent architectures as the main time-advancing scheme within the latent space, which, unlike direct approaches, leverage the sequential nature of the time-dependent reduced state. More broadly speaking, recurrent neural networks (RNNs) have found extensive application in high-dimensional dynamical systems modeling [100, 102] and in the construction of DL-based ROMs for time-dependent problems [21, 97]. RNN-based techniques have been first coupled with traditional linear dimensionality reduction techniques, adopting long short-term memory (LSTM) [40, 54] units to model the reduced state evolution obtained by means of POD [57, 67, 105]. Considering AE-based architectures, [43] proposes a recurrent convolutional AE coupled with LSTMs to model the latent state evolution. Among hybrid approaches, leveraging both POD and AEs in a recurrent setting, we cite the POD-LSTM-ROM architecture [34]. Concerning fluid flow modeling applications, LSTMs units are also adopted in [108], where a temporal context, encoding the current state and a series of previous states, is leveraged by the RNN to produce a latent sequence which is decoded back to the high-dimensional space. Similarly, the framework of *learning the effective dynamics* (LED), proposed in [99, 101], relies on an AE-LSTM architecture and adopts an end-to-end formulation to simultaneously learn the low-dimensional latent representation and its dynamics, rather than using a two-stage approach. This framework has been further developed in [61], incorporating uncertainty quantification techniques and leveraging continual learning. Regarding autoregressive approaches, [109] proposes the *latent evolution of partial differential equations* (LE-PDE) framework, which adopts a latent time-advancing scheme based on a residual update. This approach employs both a dynamic and a static encoder, to respectively reduce the full-order state dimensionality and embed system parameters, allowing for a more flexible way of handling parametric information. Similarly, [72] adopts a residual update to learn a latent convolutional propagator.

Dynamically-motivated approaches. The inherent dynamical structure of the FOM has inspired the development of approaches that leverage ODE-based formulations and physics inductive biases in order to capture the underlying dynamics of the modeled problem. The sparse identification of nonlinear dynamics (SINDy) [13] framework has been adopted in the context of latent dynamics modeling [14, 37], allowing to perform latent system identification via sparse regression on a set of candidate basis functions. While ensuring an interpretable representation of the latent dynamics, such a method faces difficulties in the parameterized context, lacking a proper parameterization mechanism. To address this limitation, [8, 36] propose to identify a set of local models, each one covering a sub-region of the parameter space, whose combination allows guaranteeing a specified accuracy level across the entire parameter space. A parameterized extension of the AE-SINDy method is presented in [20], in which the reduced dynamics is represented via a parameterized ODE system and continuation algorithms are adopted to track periodic responses' evolution. Other approaches include *dynamic mode decomposition* (DMD) [2, 94], and models based on *Koopman operator* (KO) theory [12, 64, 81]. In particular, in the multi-query setting, [27] proposes a DMD-based approach to model parameterized latent dynamics by fitting multiple latent models for different parameter instances, and then relying on interpolation during the online stage. A variation of [99] based on KO and the Mori-Zwanzig formalism is proposed in [76].

Implicit dynamics learning. At the intersection of deep learning and differential equations [28], the recently introduced concept of *neural ODEs* (NODEs) [16, 58] has enabled a novel paradigm in modeling time-dependent data [55, 92], departing from traditional discrete-time approaches. NODEs' ability to learn continuously-evolving representations, by implicitly learning the vector field parameterizing the

underlying dynamics, has bridged the gap between DNNs and continuous dynamical systems [4, 26, 73]. This has motivated their adoption in the context of reduced order modeling for time-dependent nonlinear problems [25, 29], by coupling NODEs with AEs, leading to the AE-NODE architecture. Such a paradigm has been adapted to physics-informed settings by [95] to address data-scarce scenarios. Specifically, in the context of multi-query reduced order modeling, [70] extended the concept of NODEs to include the parametric information, leading to the concept of *parameterized neural ODEs* (PNODEs), resulting in improved generalization capabilities in a parameterized context and extending the applicability of AE-(P)NODE-based ROMs in a multi-query setting. Recently, NODEs-based latent dynamics modeling has been extended towards a spatially-continuous formulation via implicit neural representation (INR)-based decoders [107, 110].

2 Towards a time-continuous ROM approximation

In this section, we introduce the problem we are interested in, and review the construction of a ROM by providing a brief overview on the main steps involved, that is, the computation of a suitable trial manifold and the identification of a reduced (latent) dynamics on it.

2.1 Problem formulation

Our focus is on the solution of time-dependent nonlinear parameterized PDEs, whose parametric dependence is characterized by a set of parameters $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^{n_\mu}$, with $\mathcal{P}$ compact, representing system configurations, physical or geometrical-related properties, boundary conditions and/or initial conditions. To ensure consistency with the formulation of the methods proposed later, we adopt an algebraic approach to the problem. Indeed, by means of numerical methods (e.g., finite element method, spectral element method), the semi-discretized high-fidelity problem consists of a high-dimensional dynamical system, which we refer to as full-order model (FOM). Given an instance of the parameters $\boldsymbol{\mu} \in \mathcal{P}$, we thus consider the following initial value problem (IVP)

$$\begin{cases} \dot{\mathbf{u}}_h(t;\boldsymbol{\mu}) = \mathbf{f}_h(t, \mathbf{u}_h(t;\boldsymbol{\mu});\boldsymbol{\mu}), \quad t \in (t_0, T], \\ \mathbf{u}_h(t_0;\boldsymbol{\mu}) = \mathbf{u}_{0,h}(\boldsymbol{\mu}), \end{cases} \tag{2.1}$$

where $\mathbf{u}_h : [t_0, T] \times \mathcal{P} \to \mathbb{R}^{N_h}$ denotes the high-fidelity parameterized solution of (2.1) and $\mathbf{u}_{0,h} : \mathcal{P} \to \mathbb{R}^{N_h}$ the initial value. The nonlinear function $\mathbf{f}_h : (t_0, T] \times \mathbb{R}^{N_h} \times \mathcal{P} \to \mathbb{R}^{N_h}$ defines the parameterized dynamics of the system. Thus, the aim of reduced order modeling techniques is to provide an approximation of the *solution manifold*

$$\mathcal{S}_h = \{\mathbf{u}_h(t;\boldsymbol{\mu}) : t \in [t_0, T], \boldsymbol{\mu} \in \mathcal{P}\} \subset \mathbb{R}^{N_h},$$

that is, the set of solutions of (2.1) in a suitable time interval $t \in [t_0, T]$ and for a prescribed set of parameters $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^{n_\mu}$.

Solution manifold. The approximation is performed through the construction of the so-called *reduced trial manifold*, denoted by $\tilde{\mathcal{S}}_h^n$, which can be either linear or nonlinear, depending on the chosen dimensionality reduction technique. In both linear and nonlinear cases, the fundamental assumption is that the intrinsic dimensionality, denoted by n, of the solution manifold $\mathcal{S}_h$ is significantly smaller than the FOM dimension N_h, i.e. $n \ll N_h$, formally stated as the *manifold hypothesis*. Such assumption, in the general nonlinear case, translates into the existence of a nonlinear map $\Psi' : \mathbb{R}^n \to \mathbb{R}^{N_h}$ such that

$$\mathbf{u}_n(t;\boldsymbol{\mu}) \mapsto \Psi'(\mathbf{u}_n(t;\boldsymbol{\mu})) = \tilde{\mathbf{u}}_h(t;\boldsymbol{\mu}) \approx \mathbf{u}_h(t;\boldsymbol{\mu}), \tag{2.2}$$

where $\mathbf{u}_n(t;\boldsymbol{\mu}) : [t_0, T] \times \mathcal{P} \to \mathbb{R}^n$ are the ROM intrinsic coordinates, or latent variables, describing the dynamics of the system onto the low-dimensional trial manifold. Thus, the reduced *nonlinear* trial manifold can be characterized as follows

$$\mathcal{S}_h \approx \tilde{\mathcal{S}}_h^n = \{\Psi'(\mathbf{u}_n(t;\boldsymbol{\mu})) : \mathbf{u}_n(t;\boldsymbol{\mu}) \in \mathbb{R}^n, t \in [t_0, T], \boldsymbol{\mu} \in \mathcal{P}\} \subset \mathbb{R}^{N_h},$$

and the associated dimensionality reduction problem takes the form

$$\min_{\Psi', \Psi} \|\mathbf{u}_h - \Psi' \circ \Psi(\mathbf{u}_h)\|^2_{W^{0,2}([t_0,T]\times\mathcal{P};\mathbb{R}^{N_h})}. \tag{2.3}$$

The adoption of nonlinear dimensionality reduction paradigms has been mainly driven by the advancements in the field of DL [69], where the nonlinear maps Ψ, Ψ', referred to as *encoder* and *decoder*, respectively, are parameterized by NNs. Their composition $\Psi' \circ \Psi : \mathbb{R}^{N_h} \to \mathbb{R}^{N_h}$ approximating the identity Id_{N_h}, is referred to as *autoencoder* (AE) [53], and the reduced state $\mathbf{u}_n(t;\boldsymbol{\mu}) = \Psi(\mathbf{u}_h(t;\boldsymbol{\mu})) \in \mathbb{R}^n$ is known as the *latent* state. Moreover, we highlight that, in the linear case, the map (2.2) is represented by a matrix $V \in \mathbb{R}^{N_h \times n}$, which leads to projection-based methods.

Reduced dynamics. The second main step in the construction of a ROM is represented by the identification of the reduced dynamics on the reduced trial manifold. In this regard, multiple strategies can be adopted. In a classical projection-based ROM, the time-evolution of the reduced dynamics is modeled by means of a parameterized dynamical system of dimension n. This involves replacing the high-fidelity solution $\mathbf{u}_h(t;\boldsymbol{\mu})$ in (2.1) with the projection-based approximation $V\mathbf{u}_n(t;\boldsymbol{\mu})$, and imposing that the residual associated to the first equation of (2.1) is orthogonal to the n-dimensional subspace spanned by the columns of the matrix V, thus obtaining

$$\begin{cases} \dot{\mathbf{u}}_n(t;\boldsymbol{\mu}) = V^T \mathbf{f}_h(t, V\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}), \quad t \in (t_0, T], \\ \mathbf{u}_n(t_0;\boldsymbol{\mu}) = V^T \mathbf{u}_{0,h}(\boldsymbol{\mu}). \end{cases} \tag{2.4}$$

The three main bottlenecks often arising with projection-based ROMs are: *(i)* the high dimension $n \gg n_{\boldsymbol{\mu}}+1$ of the low-dimensional subspace, much larger than the intrinsic dimension of the solution manifold, *(ii)* the need to rely on hyper-reduction techniques to assemble the operators appearing in the ROM (2.4) in order not to rely on expensive N_h-dimensional arrays [15], and *(iii)* their intrusive nature, due to the fact that the dynamics $\mathbf{f}_h$ must be known in order to construct the ROM.
More recently, in order to overcome these critical issues, regression-based approaches have been adopted, by modeling the relationship $(t, \boldsymbol{\mu}) \mapsto \mathbf{u}_n(t;\boldsymbol{\mu})$ via a mapping $\phi : [t_0, T] \times \mathcal{P} \to \mathbb{R}^n$. Such strategy has been employed in multiple DL-based reduced order modeling approaches [7, 33, 52, 103]; indeed, due to its regression-based formulation, it avoids the solution of the reduced dynamical system (2.4) by directly modeling the relationship between the parameters and the latent variables. However, we emphasize that, in contrast to classical projection-based ROMs, regression-based techniques lose the IVP structure of the problem. Moreover, another issue of such approaches is represented by their poor performance at time-extrapolation tasks and interpolating the solution at time instances in between the discrete training data. This has motivated the adoption of recurrent-based approaches to model the reduced dynamics, which partially solve the previous issues [34, 104].

2.2 Time-continuous approximation

In the reduced order modeling context, most of the attention, at least up to now, has been placed on trying to decrease as much as possible the spatial complexity. Indeed, less attention has been focused on the modeling and treatment of the temporal evolution. Motivated by this, we aim at developing a framework capable of generating ROMs that capture the underlying temporal dynamics of the modeled system, in the sense that the approximation can be queried at any arbitrary time instance by retaining a sufficient accuracy. This property, which is not granted a priori [66, 89], refers to the capability of the ROM to represent the continuous dynamics of a given system – that is, our model shall approximate a continuous differential operator, rather than only discrete points.
To enhance generalization capabilities at time interpolation and extrapolation tasks, we have identified the following key properties:

1. *Zero-shot time-resolution invariance.* The approximation error between the FOM and ROM solutions is almost constant as the time discretization is refined, even if the ROM is constructed using data sampled at a specific time-resolution;

2. *Causality.* The approximation at the next step depends on the previous steps;

3. *Initial value problem (IVP) structure.* The ROM preserves the FOM structure and only the initial value is needed to predict the time-evolution.

Several ROM techniques are fundamentally tied to the time discretization employed during the offline training stage. This dependency on time discretization carries over to recent DL-based approaches as well. As a result, these models eventually require rebuilding, additional training or fine-tuning stages, if they need to be tested on novel temporal discretizations, in contrast with feature 1. Causality is required at both offline training and online testing times, and it could be achieved, for instance, by means of recurrent architectures. Property 3 is deeply motivated by the IVP-structure of the FOM, that is, the aim is to model PDEs' solution evolution by means of a model architecture which has an IVP-like structure, thus only the initial value is needed in order to predict the time evolution. So, the desired framework is different from other DL-based surrogate models employed in the literature in the context of evolutionary problems, usually recurrent or autoregressive-based, typically relying on an input context window of multiple previous high-fidelity snapshots (look-back) to produce the next step. From a computational perspective, this aspect compromises the full independence of the ROM from the FOM in terms of online costs, potentially limiting the efficiency of the resulting ROM. Moreover, autoregressive approaches, despite showing short-term accuracy, often lack long-term stability, requiring ad hoc methods to promote stable long roll-outs [9, 74].
The previous three concepts led us to the definition of a *time-continuous approximation* property, which is an extension of the one provided in [66]. Properties 2 and 3 affect the NN structure of the ROM solution depending only on the initial datum, the previous time instances, and the physical parameter $\boldsymbol{\mu}$, while property 1 results in a practical criterion to assess the generalization capabilities of the approximation with respect to the temporal discretization.

Definition 2.1 (Time-continuous approximation)**.** Let $[t_0, T] \subset \mathbb{R}_+$ be a generic time interval and $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^{n_\mu}$. Let us denote by $\mathcal{T}_{\Delta t} = \{t_i\}_{i=0}^{N_{\Delta t}}$ a partition of $[t_0, T]$, with $t_i = t_0 + i\Delta t$ and $\Delta t > 0$ a chosen time step. Suppose that the ROM surrogate $\hat{\mathbf{u}}_h^i(\boldsymbol{\mu}) = \boldsymbol{\Theta}(\mathbf{u}_{0,h}, \{t_j\}_{j=0}^i; \boldsymbol{\mu}) \approx \mathbf{u}_h(t_i; \boldsymbol{\mu})$ is trained over data generated with a time discretization $\mathcal{T}_{\Delta t^*} = \{t_k\}_{k=0}^{N_{\Delta t^*}}$, $t_k = t_0 + k\Delta t^*$, with time step $\Delta t^* > 0$. Then, the ROM solution is said to be a *time-continuous approximation* if, at prediction time, for every time-step $\Delta t \leq \Delta t^*$, it holds

$$\sup_{t_i \in \mathcal{T}_{\Delta t}} ||\mathbf{u}_h(t_i; \boldsymbol{\mu}) - \hat{\mathbf{u}}_h^i(\boldsymbol{\mu})|| \lessapprox \sup_{t_k \in \mathcal{T}_{\Delta t^*}} ||\mathbf{u}_h(t_k; \boldsymbol{\mu}) - \hat{\mathbf{u}}_h^k(\boldsymbol{\mu})|| \quad \forall \boldsymbol{\mu} \in \mathcal{P}. \tag{2.5}$$

In other words, we are asking that the maximum error over any time discretization, employing $\Delta t < \Delta t^*$, is of the same order as the maximum error over the training time discretization, that is, the training accuracy is preserved on finer time discretizations employed during testing. Such property highlights that the model has time generalization capabilities, meaning that the learned dynamics meaningfully captures the underlying phenomena, at the high-dimensional level, rather than simply fitting the discrete data points. In the following, we aim at showing that by means of latent dynamics models, it is possible to ensure the fulfillment of the time-continuous approximation property at the high-dimensional level, that is, to obtain a time-continuous ROM approximation $\hat{\mathbf{u}}_h(t; \boldsymbol{\mu}) \in \mathbb{R}^{N_h}$.

3 Modeling of the latent dynamics

In the following, we formalize the concept of *latent dynamics model* (LDM), in the context of dimensionality reduction techniques for parameterized nonlinear dynamical systems. Our abstract setting allows to understand the temporal behavior of such modeling framework, starting from the time-continuous setting, which can be assimilated to the case of infinite time observations, with the end goal of deriving suitable error bound and stability estimate for the approximated solution. Developing a learnable framework operating in a time-continuous context represents a crucial step to *(i)* remove the dependence on the time-step employed at training time, *(ii)* be able to deal with irregularly spaced time grids, and *(iii)* enhance long-term time-extrapolation capabilities beyond the training time horizon. In particular, such modeling framework would allow us to generalize to temporal grids much finer than the one employed during the offline stage, thus removing the need of rebuilding the ROM, redefining the model architecture,

or requiring additional training and fine-tuning on the novel discretizations. The time-continuous setting will serve as starting point for the introduction of numerical integration schemes to solve the temporal dynamics in a space of reduced dimensions, while showing that the resulting time-discrete framework satisfies classical properties of time schemes for the solution of dynamical systems. Finally, we present a learnable setting where the different components are approximated by means of NNs. In particular, we aim at studying the approximation properties of the ROM solution, by showing that the approximation error between the ROM and the FOM solutions is bounded, while addressing its time-continuous approximation capability.

3.1 Latent dynamics models

In nonlinear dimensionality reduction, a key challenge, regardless of the nature of the maps adopted in the dimensionality reduction technique, lies in modeling the dynamics of the reduced (latent) state $\mathbf{u}_n(t;\boldsymbol{\mu})$. Characterizing the problem in a time interval $[t_0, T]$, and assuming a sufficient regularity of the latent state, that is, $\mathbf{u}_n \in C^1([t_0,T]\times\mathcal{P};\mathbb{R}^n)$, the latter task entails the discovery of $\mathbf{f}_n : (t_0,T]\times\mathbb{R}^n\times\mathcal{P}\to\mathbb{R}^n$ encoding the latent state dynamics, so that

$$\dot{\mathbf{u}}_n(t;\boldsymbol{\mu}) = \mathbf{f}_n(t,\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}). \tag{3.1}$$

Taking advantage of (3.1), in the following we properly address the problem of characterizing both the nonlinear dimensionality reduction and the identification of the latent dynamics. We first provide a formal definition of *latent dynamics problem* (LDP), and we proceed by introducing the concept of *latent dynamics model* (LDM), defining the mathematical context of time-continuous dimensionality reduction in a general setting.

Definition 3.1 (Latent dynamics problem)**.** Let $\mathbf{u}_h : [t_0,T]\times\mathcal{P}\to\mathcal{S}_h\subset\mathbb{R}^{N_h}$ be a high-dimensional parameterized time-dependent state, such that $\mathbf{u}_h\in C^1([t_0,T]\times\mathcal{P};\mathbb{R}^n)$, evolving accordingly to the first-order IVP

$$\begin{cases} \dot{\mathbf{u}}_h(t;\boldsymbol{\mu}) = \mathbf{f}_h(t,\mathbf{u}_h(t;\boldsymbol{\mu});\boldsymbol{\mu}), & t\in(t_0,T],\\ \mathbf{u}_h(t_0;\boldsymbol{\mu}) = \mathbf{u}_{0,h}(\boldsymbol{\mu}), \end{cases} \tag{3.2}$$

with $\mathbf{f}_h : (t_0,T]\times\mathbb{R}^{N_h}\times\mathcal{P}\supset D_h\to\mathbb{R}^{N_h}$ the high-dimensional nonlinear vector field describing its dynamics, for each $\boldsymbol{\mu}\in\mathcal{P}$. Let $\mathbf{u}_n : [t_0,T]\times\mathcal{P}\to\mathcal{S}_n\subset\mathbb{R}^n$ be a low-dimensional parameterized time-dependent state ($n\ll N_h$), such that $\mathbf{u}_n\in C^1([t_0,T]\times\mathcal{P};\mathbb{R}^n)$. Then, the low-dimensional state $\mathbf{u}_n(t;\boldsymbol{\mu})$ defines a latent dynamics for the high-dimensional state evolution $\mathbf{u}_h(t;\boldsymbol{\mu})$ if there exists a triple $(\Psi^*,\Psi'^*,\mathbf{f}_n^*)$ solving the following *latent dynamics problem*

$$\min_{\Psi',\mathbf{u}_n} \|\mathbf{u}_h-\Psi'(\mathbf{u}_n)\|^2_{W^{0,2}([t_0,T]\times\mathcal{P};\mathbb{R}^{N_h})} \quad \text{s.t.} \quad \begin{cases} \dot{\mathbf{u}}_n(t;\boldsymbol{\mu}) = \mathbf{f}_n(t,\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}), & t\in(t_0,T],\\ \mathbf{u}_n(t_0;\boldsymbol{\mu}) = \Psi(\mathbf{u}_{0,h}(\boldsymbol{\mu})), \end{cases} \tag{3.3}$$

with $\mathbf{f}_n : (t_0,T]\times\mathbb{R}^n\times\mathcal{P}\supset D_n\to\mathbb{R}^n$ the low-dimensional nonlinear vector field describing the dynamics, while $\Psi:\mathbb{R}^{N_h}\to\mathbb{R}^n$ and $\Psi':\mathbb{R}^n\to\mathbb{R}^{N_h}$ are Lipschitz-continuous mappings.

We remark that minimal requirements to ensure well-posedness of the high- and low-dimensional IVPs (3.2)-(3.3) on the vector fields $\mathbf{f}_h$ ($\mathbf{f}_n$), are *(i)* continuity in D_h (D_n), and *(ii)* local Lipschitz continuity in D_h (D_n) with respect to $\mathbf{u}_h$ ($\mathbf{u}_n$), uniformly in t and $\boldsymbol{\mu}$; in this way, we ensure local existence and uniqueness of the solutions, with D_h and D_n open in $(t_0,T]\times\mathbb{R}^{N_h}\times\mathcal{P}$ and $(t_0,T]\times\mathbb{R}^n\times\mathcal{P}$, respectively. Additionally, boundedness of $\mathbf{f}_h$ and $\mathbf{f}_n$ in $\bar{D}_h$ and $\bar{D}_n$, respectively, ensures global existence of the solutions of (3.2)-(3.3).

Definition 3.2 (Latent dynamics model)**.** The triple $(\Psi,\Psi',\mathbf{f}_n)$ satisfying (3.3) is called *latent dynamics model* for $\mathbf{u}_h(t;\boldsymbol{\mu})$ and $\tilde{\mathbf{u}}_h(t;\boldsymbol{\mu})$ is the resulting approximation, reading in an explicit

form as

$$\mathbf{u}_h(t;\boldsymbol{\mu}) \approx \tilde{\mathbf{u}}_h(t;\boldsymbol{\mu}) \coloneqq \Psi'\left(\Psi(\mathbf{u}_{0,h}(\boldsymbol{\mu})) + \int_{t_0}^{t} \mathbf{f}_n(\tau, \mathbf{u}_n(\tau;\boldsymbol{\mu});\boldsymbol{\mu})d\tau\right). \tag{3.4}$$

Let us discuss the regularity properties introduced in Definition 3.1. Specifically, the smoothness assumptions on the high-dimensional parameterized state $\mathbf{u}_h$ and its low-dimensional *latent* counterpart $\mathbf{u}_n$, belonging to a functional space of the form $C^1([t_0,T]\times\mathcal{P};\mathbb{R}^*)$, entail two key properties. Firstly, it ensures that the states are continuously differentiable with respect to the temporal variable over $[t_0,T]$, secondly that the parameter-to-solution map $(t,\boldsymbol{\mu}) \mapsto \mathbf{u}_n(t;\boldsymbol{\mu})$ is Lipschitz-regular. Moreover, we highlight that the LDP is formulated as a joint minimization problem with respect to both the maps Ψ, Ψ' and the latent dynamics $\mathbf{f}_n$, where the latter play a central role in the dimensionality reduction aspect of the proposed framework. In contrast to DL-ROM paradigms [33], the role of Ψ is to map only the high-dimensional initial value $\mathbf{u}_{0,h} \in \mathbb{R}^{N_h}$ to $\Psi(\mathbf{u}_{0,h}) \in \mathbb{R}^n$ as the dynamics is recovered through the evolution of the latent state, via the integration of $\mathbf{f}_n$. On the other hand, the mapping Ψ' acts at any time instance, allowing to project the latent state evolution $\mathbf{u}_n(t;\boldsymbol{\mu})$ back to the high-dimensional space. We refer to Figure 2 for a visualization of the overall dimensionality reduction flow. Nonetheless, we emphasize that the choice of the latent dimension n is essential to calibrate the accuracy of LDMs; in the following, we present a framework to suitably address this matter.

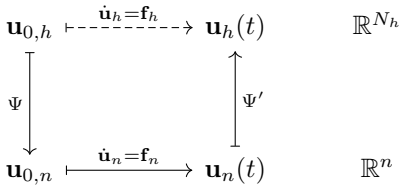

Figure 2: Commutative diagram explaining the LDM scheme, approximating the mapping $\mathbf{u}_{0,h} \mapsto \mathbf{u}_h(t)$ defined by the full-order dynamics $\mathbf{f}_h$, by means of a latent dynamics $\mathbf{f}_n$ and the nonlinear projection and reconstruction maps Ψ, Ψ'.

3.1.1 Nonlinear dimensionality reduction in a dynamical setting

In linear dimensionality reduction, the selection of the reduced dimension n is determined by the decay of the singular values of the snapshot matrix [86], while the characterization of n in a nonlinear setting requires a different theoretical environment. Specifically, here, we aim at characterizing the nonlinear dimensionality reduction framework in a dynamical setting, starting from the LDM formulation. To do that, we first provide a preliminary result that unveils the contribution of Ψ, Ψ' and $\mathbf{f}_n$ to the approximation error, namely,

Proposition 3.3. *For any $\boldsymbol{\mu} \in \mathcal{P}$, under the hypotheses of Section 3.1, it holds that*

$$\begin{aligned}\|\mathbf{u}_h(t;\boldsymbol{\mu}) - \tilde{\mathbf{u}}_h(t;\boldsymbol{\mu})\| \leq\ & \|\mathbf{u}_{0,h} - \Psi' \circ \Psi(\mathbf{u}_{0,h})\|_{W^{0,\infty}(\mathcal{P};\mathbb{R}^{N_h})} \\ & + |T-t_0| \|\mathbf{f}_h(\mathbf{u}_h) - \mathbf{J}_{\Psi',\mathbf{u}_n}\mathbf{f}_n(\mathbf{u}_n))\|_{W^{0,\infty}([t_0,T]\times\mathcal{P};\mathbb{R}^{N_h})},\end{aligned}$$

for any $t \in [t_0,T]$, where $\mathbf{J}_{\Psi',\mathbf{u}_n}$ is the Jacobian of Ψ' with respect to $\mathbf{u}_n$.

Proof. Considering the LDM formulation over $[t_0,T]$, namely

$$\tilde{\mathbf{u}}_h(t;\boldsymbol{\mu}) = \Psi'\left(\Psi(\mathbf{u}_{0,h}(\boldsymbol{\mu})) - \int_{t_0}^{t} \mathbf{f}_n(s, \mathbf{u}_n(s;\boldsymbol{\mu});\boldsymbol{\mu})ds\right) = \Psi'(\mathbf{u}_n(t;\boldsymbol{\mu})),$$

we observe that, thanks to the chain rule, the following IVP is satisfied

$$\begin{cases} \partial_t \tilde{\mathbf{u}}_h(t;\boldsymbol{\mu}) = \mathbf{J}_{\Psi',\mathbf{u}_n}(t;\boldsymbol{\mu})\partial_t \mathbf{u}_n(t;\boldsymbol{\mu}) = \mathbf{J}_{\Psi',\mathbf{u}_n}(t;\boldsymbol{\mu})\mathbf{f}_n(t,\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}), \quad t \in (t_0,T], \\ \tilde{\mathbf{u}}_h(0;\boldsymbol{\mu}) = \Psi' \circ \Psi(\mathbf{u}_{0,h}(\boldsymbol{\mu})). \end{cases}$$

Thus, it is trivial to conclude

$$\begin{aligned} &\|\mathbf{u}_h(t;\boldsymbol{\mu}) - \tilde{\mathbf{u}}_h(t;\boldsymbol{\mu})\| \\ &\quad = \left\| \mathbf{u}_{0,h}(\boldsymbol{\mu}) + \int_{t_0}^t \mathbf{f}_h(s,\mathbf{u}_h(s;\boldsymbol{\mu});\boldsymbol{\mu})ds - \Psi' \circ \Psi(\mathbf{u}_{0,h}(\boldsymbol{\mu})) - \int_{t_0}^t \mathbf{J}_{\Psi',\mathbf{u}_n}(s;\boldsymbol{\mu})\mathbf{f}_n(s,\mathbf{u}_n(s;\boldsymbol{\mu});\boldsymbol{\mu})ds \right\| \\ &\quad \leq \|\mathbf{u}_{0,h}(\boldsymbol{\mu}) - \Psi' \circ \Psi(\mathbf{u}_{0,h}(\boldsymbol{\mu}))\| + \left\| \int_{t_0}^t [\mathbf{f}_h(s,\mathbf{u}_h(s;\boldsymbol{\mu});\boldsymbol{\mu}) - \mathbf{J}_{\Psi',\mathbf{u}_n}(s;\boldsymbol{\mu})\mathbf{f}_n(s,\mathbf{u}_n(s;\boldsymbol{\mu});\boldsymbol{\mu})]ds \right\| \\ &\quad \leq \|\mathbf{u}_{0,h} - \Psi' \circ \Psi(\mathbf{u}_{0,h})\|_{W^{0,\infty}(\mathcal{P};\mathbb{R}^{N_h})} + |T - t_0| \|\mathbf{f}_h(\mathbf{u}_h) - \mathbf{J}_{\Psi',\mathbf{u}_n}\mathbf{f}_n(\mathbf{u}_n))\|_{W^{0,\infty}([t_0,T]\times\mathcal{P};\mathbb{R}^{N_h})}. \end{aligned}$$

□

The latter proposition reveals the deep connection between the maps Ψ, Ψ' involved in the dimensionality reduction of the full order state $\mathbf{u}_h(t;\boldsymbol{\mu})$ and the latent dynamics $\mathbf{f}_n(t,\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})$. In particular, the first term in the upper bound is related to the capability of the encoder and the decoder to reconstruct the initial condition. On the other hand, the second term depends on Ψ' and $\mathbf{f}_n$, and describes how accurate is the latent state dynamics approximation of the exact high-fidelity dynamics. We emphasize that as the latent dimension n increases, we expect that both terms converge quickly to 0. To characterize such convergence, in the wake of other works concerning the approximation capabilities of AE-based architectures [10, 30], we introduce the perfect embedding assumption, specifically crafted for latent dynamics modeling. In particular, we argue that the true FOM dynamics can be exactly represented through a sufficiently rich LDM, according to the following definition.

Definition 3.4. (Perfect embedding assumption) There exists $n^* \ll N_h$ such that, for any $n \geq n^*$, there exist

- a decoder function $\Psi' \in C^1(\mathbb{R}^n;\mathbb{R}^{N_h})$,
- a latent dynamics function $\mathbf{f}_n(t,\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})$, which is Lipschitz continuous with respect to $\mathbf{u}_n$, uniformly in t, and Lipschitz continuous with respect to t,

such that

$$\mathbf{u}_h(t;\boldsymbol{\mu}) = \Psi'(\mathbf{u}_n(t;\boldsymbol{\mu})) \quad \text{and} \quad \dot{\mathbf{u}}_n(t;\boldsymbol{\mu}) = \mathbf{f}_n(t,\mathbf{u}_n(t;\boldsymbol{\mu})) \quad \forall t \in (t_0,T]. \tag{3.5}$$

Moreover, for any $n \geq n^*$ there exists $\Psi \in C^1(\mathbb{R}^{N_h};\mathbb{R}^n)$ such that $\mathbf{u}_n(t_0;\boldsymbol{\mu}) = \Psi(\mathbf{u}_h(t_0;\boldsymbol{\mu}))$. The previous assumptions hold for any $\boldsymbol{\mu} \in \mathcal{P}$.

The purpose of the perfect embedding assumption is both to provide a suitable heuristic criterion to determine the latent dimension n and the regularity of Ψ', Ψ and $\mathbf{f}_n$. We refer the interested reader to [30] for more details on the characterization of the latent dimension in nonlinear dimensionality reduction. Moreover, we highlight a deep connection between the error decomposition formula of Proposition 3.3 and the Assumption 3.4; indeed, it is straightforward to see that if the perfect embedding assumption is attained, then both terms in the upper bound of Proposition 3.3 vanish. Finally, we remark that Assumption 3.4 will play a crucial role in the subsequent sections of this work, especially for the characterization of the time-continuous approximation property. Before delving into that matter, we first focus on the properties and numerical analysis of LDMs.

3.1.2 Stability

The dynamical nature of the proposed framework allows translating concepts strictly related to ODEs and dynamical systems to the LDM context, allowing to characterize its time-continuous approximation properties, such as *stability*. Specifically, LDMs' integral formulation (3.4) defines a parameterized

problem of the form

$$\begin{cases} \tilde{\mathbf{u}}_h(t;\boldsymbol{\mu}) = \Psi'(\mathbf{u}_n(t;\boldsymbol{\mu})), & t \in (t_0, T], \\ \dot{\mathbf{u}}_n(t;\boldsymbol{\mu}) = \mathbf{f}_n(t, \mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}), & t \in (t_0, T], \\ \mathbf{u}_n(t_0;\boldsymbol{\mu}) = \Psi(\mathbf{u}_{0,h}(\boldsymbol{\mu})), \end{cases} \tag{3.6}$$

where the latter two equations define a *latent* IVP in $\mathcal{S}_n \subset \mathbb{R}^n$, while the former defines the mapping of the latent state $\mathbf{u}_n(t;\boldsymbol{\mu})$ to the high-dimensional space $\mathcal{S}_h \subset \mathbb{R}^{N_h}$, producing an approximation $\tilde{\mathbf{u}}_h(t;\boldsymbol{\mu})$ of the FOM state. In order to analyze the stability of (3.6) in the sense of Lyapunov [3, 87], we consider the following perturbed problem

$$\begin{cases} \tilde{\mathbf{z}}_h(t;\boldsymbol{\mu}) = \Psi'(\mathbf{z}_n(t;\boldsymbol{\mu})), & t \in (t_0, T], \\ \dot{\mathbf{z}}_n(t;\boldsymbol{\mu}) = \mathbf{f}_n(t, \mathbf{z}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}) + \boldsymbol{\delta}(t), & t \in (t_0, T], \\ \mathbf{z}_n(t_0;\boldsymbol{\mu}) = \Psi(\mathbf{u}_{0,h} + \boldsymbol{\delta}_0), \end{cases} \tag{3.7}$$

denoting by $(\boldsymbol{\delta}_0, \boldsymbol{\delta}(t)) \in \mathbb{R}^{N_h} \times \mathbb{R}^n$ two perturbations such that $\|\boldsymbol{\delta}_0\| < \varepsilon,\ \|\boldsymbol{\delta}(t)\| < \varepsilon,\ \forall t \in (t_0, T]$. In particular, by setting $\varepsilon > 0$ such that the latent IVP is well-defined, we aim to show the existence of a constant $C > 0$, independent of ε, such that the following bound on the high-dimensional solutions holds:

$$\|\tilde{\mathbf{z}}_h(t;\boldsymbol{\mu}) - \tilde{\mathbf{u}}_h(t;\boldsymbol{\mu})\| < C\varepsilon, \qquad \forall t \in (t_0, T].$$

The goal is to assess the sensitivity of $\tilde{\mathbf{u}}_h(t;\boldsymbol{\mu})$ with respect to the introduced perturbations, and to unveil the role of the maps Ψ and Ψ' in guaranteeing stability, together with the appropriate requirements. We proceed by fixing $\boldsymbol{\mu} \in \mathcal{P}$, and by considering

$$\|\tilde{\mathbf{z}}_h(t) - \tilde{\mathbf{u}}_h(t)\| = \|\Psi'(\mathbf{z}_n(t)) - \Psi'(\mathbf{u}_n(t))\| \leq Lip(\Psi')\|\mathbf{z}_n(t) - \mathbf{u}_n(t)\| = Lip(\Psi')\|\boldsymbol{\xi}(t)\|, \tag{3.8}$$

having set $\boldsymbol{\xi}(t) = \mathbf{z}_n(t) - \mathbf{u}_n(t)$ and denoting by $Lip(\cdot)$ the Lipschitz constant of a given map. By differentiating, so that $\dot{\boldsymbol{\xi}}(t) = \mathbf{f}_n(t, \mathbf{z}_n(t)) - \mathbf{f}_n(t, \mathbf{u}_n(t)) + \boldsymbol{\delta}(t)$, and then integrating on (t_0, t), with $t \in (t_0, T]$, it follows that

$$\boldsymbol{\xi}(t) = \mathbf{z}_n(t_0) - \mathbf{u}_n(t_0) + \int_{t_0}^{t} \Big(\mathbf{f}_n(\tau, \mathbf{z}_n(\tau)) - \mathbf{f}_n(\tau, \mathbf{u}_n(\tau))\Big)d\tau + \int_{t_0}^{t} \boldsymbol{\delta}(\tau)d\tau.$$

Then,

$$\begin{aligned} \|\boldsymbol{\xi}(t)\| &\leq \|\mathbf{z}_n(t_0) - \mathbf{u}_n(t_0)\| + Lip(\mathbf{f}_n)\int_{t_0}^{t} \|\boldsymbol{\xi}(\tau)\|d\tau + \int_{t_0}^{t} \|\boldsymbol{\delta}(\tau)\|d\tau \\ &= \|\Psi(\mathbf{u}_h(t_0) + \boldsymbol{\delta}_0) - \Psi(\mathbf{u}_h(t_0))\| + Lip(\mathbf{f}_n)\int_{t_0}^{t} \|\boldsymbol{\xi}(\tau)\|d\tau + \int_{t_0}^{t} \|\boldsymbol{\delta}(\tau)\|d\tau \\ &\leq Lip(\Psi)\|\boldsymbol{\delta}_0\| + Lip(\mathbf{f}_n)\int_{t_0}^{t} \|\boldsymbol{\xi}(\tau)\|d\tau + \int_{t_0}^{t} \|\boldsymbol{\delta}(\tau)\|d\tau \\ &\leq (Lip(\Psi) + |t - t_0|)\varepsilon + Lip(\mathbf{f}_n)\int_{t_0}^{t} \|\boldsymbol{\xi}(\tau)\|d\tau \leq (Lip(\Psi) + |t - t_0|)\varepsilon e^{Lip(\mathbf{f}_n)|t-t_0|} \end{aligned} \tag{3.9}$$

where (3.9) follows from Gronwall's Lemma. Finally, considering (3.8), we obtain

$$\begin{aligned} \|\tilde{\mathbf{z}}_h(t) - \tilde{\mathbf{u}}_h(t)\| &\leq Lip(\Psi')(Lip(\Psi) + |t - t_0|)\varepsilon e^{Lip(\mathbf{f}_n)|t-t_0|} \qquad \forall t \in (t_0, T] \\ &\leq Lip(\Psi')(Lip(\Psi) + |T - t_0|)\varepsilon e^{Lip(\mathbf{f}_n)|T-t_0|} \end{aligned}$$

Thus, setting

$$C = C(\Psi, \Psi', \mathbf{f}_n, |T - t_0|) = Lip(\Psi')(Lip(\Psi) + |T - t_0|)e^{Lip(\mathbf{f}_n)|T-t_0|},$$

we can write

$$\|\tilde{\mathbf{z}}_h(t) - \tilde{\mathbf{u}}_h(t)\| \leq C\varepsilon,$$

with C independent of the perturbation magnitude ε. The above analysis highlights the role of the projection maps Ψ, Ψ' in ensuring the stability of (3.6) by means of their Lipschitz continuity.

3.2 Discrete latent dynamics models

The actual construction of LDMs deeply relies on numerical integration schemes for the solution of the latent IVP (3.3). Thus, we aim to describe the notion of LDM in a time-discrete setting, by considering one-step schemes for numerically approximating the time-evolution of the latent dynamics. Opting for one-step methods may appear limiting, however this choice is supported by the possibility to easily extend the framework to DL scenarios. Indeed, in such cases, explicit RK schemes are predominantly employed due to their effectiveness in balancing training efficiency and inference performance. In particular, we assume the existence of a latent state $\mathbf{u}_n(t;\boldsymbol{\mu})$ for the high-dimensional state $\mathbf{u}_h(t;\boldsymbol{\mu})$, obtained by means of a LDM $(\Psi, \Psi', \mathbf{f}_n)$ solving the associated LDP.
Thus, by introducing a discretization of the time domain $[t_0, T]$ into $N_{\Delta t}$ intervals of width Δt, with $t_k = t_0 + k\Delta t$ for $k = 0, \dots, N_{\Delta t}$, let $\mathbf{u}_n^k(\boldsymbol{\mu})$ be the numerical approximation at time step t_k of the time-continuous latent state $\mathbf{u}_n(t_k;\boldsymbol{\mu})$. Adopting Henrici's notation [50], a single step of an explicit RK scheme for the numerical approximation of the latent dynamics, reads as

$$\mathbf{u}_n^k(\boldsymbol{\mu}) = \mathbf{u}_n^{k-1}(\boldsymbol{\mu}) + \Delta t \Phi(t_{k-1}, \mathbf{u}_n^{k-1}(\boldsymbol{\mu}); \Delta t, \mathbf{f}_n, \boldsymbol{\mu}), \qquad k = 1, \dots, N_{\Delta t}, \tag{3.10}$$

denoting by Φ the method's increment function. Thus, by inserting (3.10) into the LDM framework, we can define the discrete LDM formulation (ΔLDM), reading as

$$\begin{cases} \tilde{\mathbf{u}}_h^k(\boldsymbol{\mu}) = \Psi'(\mathbf{u}_n^k(\boldsymbol{\mu})), & k = 1, \dots, N_{\Delta t}, \\ \mathbf{u}_n^k(\boldsymbol{\mu}) = \mathbf{u}_n^{k-1}(\boldsymbol{\mu}) + \Delta t \Phi(t_{k-1}, \mathbf{u}_n^{k-1}(\boldsymbol{\mu}); \Delta t, \mathbf{f}_n, \boldsymbol{\mu}), & k = 1, \dots, N_{\Delta t}, \\ \mathbf{u}_n^0(\boldsymbol{\mu}) = \Psi(\mathbf{u}_{0,h}(\boldsymbol{\mu})). \end{cases} \tag{3.11}$$

As highlighted in $(3.11)_1$, numerically solving the latent IVP (3.3) leads to a numerical approximation $\tilde{\mathbf{u}}_h^k(\boldsymbol{\mu})$ of the time-continuous LDM high-dimensional reconstructed high-fidelity state $\tilde{\mathbf{u}}_h(t;\boldsymbol{\mu})$, thus leading to a further approximation of the high-dimensional FOM state $\mathbf{u}_h(t;\boldsymbol{\mu})$ by means of $\tilde{\mathbf{u}}_h^k(\boldsymbol{\mu})$. Thus, the error between the discrete approximation provided by ΔLDM and the FOM solution, at t_k, can be decomposed as follows

$$\|\mathbf{u}_h(t_k, \boldsymbol{\mu}) - \tilde{\mathbf{u}}_h^k(\boldsymbol{\mu})\| \leq \|\mathbf{u}_h(t_k, \boldsymbol{\mu}) - \tilde{\mathbf{u}}_h(t_k, \boldsymbol{\mu})\| + \|\tilde{\mathbf{u}}_h(t_k, \boldsymbol{\mu}) - \tilde{\mathbf{u}}_h^k(\boldsymbol{\mu})\|, \tag{3.12}$$

taking into account (i) the first stage of approximation of the full-order state by means of the LDM in time-continuous settings, and (ii) the second stage of numerical approximation, accounting for the numerical error source due to the ΔLDM approximating the LDM. In this view, it is of critical importance to characterize the concepts of *convergence, consistency* and *zero-stability* for the introduced discrete formulation (3.11), in order to better understand how the properties of the numerical scheme are transferred to the ΔLDM approximation, and to further extend the error decomposition formula (3.16) to the discrete case, by means of (3.12). In the following the parameter dependence is temporarily dropped, assuming again to fix $\boldsymbol{\mu} \in \mathcal{P}$.

Convergence. Given the ΔLDM formulation, we aim to characterize its convergence, defined as

$$\lim_{\Delta t \to 0} \|\tilde{\mathbf{u}}_h(t_k) - \tilde{\mathbf{u}}_h^k\| = 0, \quad \forall k = 0, \dots, N_{\Delta t}.$$

The convergence of the ΔLDM discrete evolution to the LDM approximation is naturally related, by construction, to the convergence of the time integration scheme employed in the latent space. In particular, assuming to employ a method of order p, the following convergence result holds for the discretized latent problem:

$$\|\mathbf{u}_n(t_k) - \mathbf{u}_n^k\| = O(\Delta t^p).$$

Thus, a convergence result for the overall framework can be built on top of the one related to the latent time-stepping scheme, by involving the reconstruction map Ψ'. Indeed, by means of Ψ' Lipschitz conti-

nuity, the order of convergence is preserved under the mapping from $\mathcal{S}_n \to \mathcal{S}_h$ for the high-dimensional reconstruction $\tilde{\mathbf{u}}_h^k$, reading as

$$\|\tilde{\mathbf{u}}_h(t_k) - \tilde{\mathbf{u}}_h^k\| = \|\Psi'(\mathbf{u}_n(t_k)) - \Psi'(\mathbf{u}_n^k)\| \leq Lip(\Psi')\|\mathbf{u}_n(t_k) - \mathbf{u}_n^k\| \leq Lip(\Psi')C\Delta t^p.$$

Consistency. Here, we aim to characterize how the consistency of the numerical method for the solution of the latent IVP, employed in the ΔLDM formulation, is transferred to the high dimensional state, investigating the relationship between the high and low-dimensional local truncation errors (LTEs). In order to define the consistency property for the method employed for discretizing the latent ODE, we consider the residual $\boldsymbol{\varepsilon}_n^k \in \mathbb{R}^n$ at time t_k, defined as

$$\boldsymbol{\varepsilon}_n^k = \mathbf{u}_n(t_k) - \mathbf{u}_n(t_{k-1}) - \Delta t\Phi(t_{k-1}, \mathbf{u}_n(t_{k-1}); \mathbf{f}_n, \Delta t).$$

Moreover, we define the residual $\boldsymbol{\varepsilon}_h^k \in \mathbb{R}^{N_h}$ that generates after one-step of the ΔLDM scheme, at t_k, with respect to the time-continuous FOM solution, by considering of having as initial condition $\mathbf{u}_h(t_k)$

$$\boldsymbol{\varepsilon}_h^k = \underbrace{\tilde{\mathbf{u}}_h(t_k)}_{\text{LDM}} - \underbrace{\Psi'(\Psi(\mathbf{u}_h(t_{k-1})) + \Delta t\Phi(t_{k-1}, \Psi(\mathbf{u}_h(t_{k-1})); \mathbf{f}_n, \Delta t))}_{\text{1-step } \Delta\text{LDM}}.$$

At this point, we relate the high- and low-dimensional residuals, in order to draw the connection between consistency of the LDM scheme and the one of the method employed in the latent space, as follows:

$$\begin{aligned}
\|\boldsymbol{\varepsilon}_h^k\| &= \|\Psi'(\mathbf{u}_n(t_k)) - \Psi'(\Psi(\mathbf{u}_h(t_{k-1})) + \Delta t\Phi(t_{k-1}, \Psi(\mathbf{u}_h(t_{k-1})); \mathbf{f}_n, \Delta t))\| \\
&\leq Lip(\Psi')\|\mathbf{u}_n(t_k) - \Psi(\mathbf{u}_h(t_{k-1})) - \Delta t\Phi(t_{k-1}, \Psi(\mathbf{u}_h(t_{k-1})); \mathbf{f}_n, \Delta t)\| \\
&= Lip(\Psi')\|\mathbf{u}_n(t_k) - \mathbf{u}_n(t_{k-1}) - \Delta t\Phi(t_{k-1}, \mathbf{u}_n(t_{k-1}); \mathbf{f}_n, \Delta t)\| \\
&= Lip(\Psi')\|\boldsymbol{\varepsilon}_n^k\|.
\end{aligned}$$

Thus, letting $\boldsymbol{\tau}_*^k(\Delta t) = \boldsymbol{\varepsilon}_*^k/\Delta t$ be the LTEs, with $*$ either h or n, the following holds

$$\|\boldsymbol{\tau}_h^k(\Delta t)\| \leq Lip(\Psi')\|\boldsymbol{\tau}_n^k(\Delta t)\|, \qquad 1 \leq k \leq N_{\Delta t},$$

which, in terms of the global truncation errors, reads as

$$\tau_h(\Delta t) \leq Lip(\Psi')\tau_n(\Delta t).$$

Thus, assuming consistency of the latent scheme, i.e. $\tau_n(\Delta t) \to 0$, as $\Delta t \to 0$, the consistency of the discretized LDM scheme is guaranteed.

Zero-stability. Assuming to employ a *zero-stable* explicit one-step numerical method for the approximation of the latent IVP, we aim to characterize how such property, belonging to the latent state approximation $\mathbf{u}_n^k$, affects the reconstructed high-fidelity approximation $\tilde{\mathbf{u}}_h^k$. Thus, we proceed by considering the perturbed ΔLDM formulation, defined as

$$\begin{cases}
\tilde{\mathbf{z}}_h^k = \Psi'(\mathbf{z}_n^k), & k = 1, \dots, N_{\Delta t}, \\
\mathbf{z}_n^k = \mathbf{z}_n^{k-1} + \Delta t\Big(\Phi(t_{k-1}, \mathbf{z}_n^{k-1}; \mathbf{f}_n, \Delta t) + \boldsymbol{\delta}_n^k\Big), & k = 1, \dots, N_{\Delta t}, \\
\mathbf{z}_n^0 = \Psi(\mathbf{z}_h^0 + \boldsymbol{\delta}_h^0),
\end{cases} \tag{3.13}$$

with $(\boldsymbol{\delta}_h^0, \boldsymbol{\delta}_n^k) \in \mathbb{R}^{N_h} \times \mathbb{R}^n$ suitable perturbations. In particular, we aim to show that $\exists \Delta t^* > 0,\ \exists \tilde{C} > 0$ such that $\forall \Delta t \leq \Delta t^*$, $\forall \varepsilon > 0$ small enough, if $\|\boldsymbol{\delta}_h^0\| \leq \varepsilon,\ \|\boldsymbol{\delta}_n^k\| \leq \varepsilon,\ 1 \leq k \leq N_{\Delta t}$, then

$$\|\tilde{\mathbf{z}}_h^k - \tilde{\mathbf{u}}_h^k\| \leq \tilde{C}\varepsilon, \qquad 1 \leq k \leq N_{\Delta t}. \tag{3.14}$$

Assuming to employ a zero-stable method for the latent IVP problem approximation, a result of the form (3.14) holds for the discretized latent state, i.e.

$$\|\mathbf{z}_n^k - \mathbf{u}_n^k\| \leq C\varepsilon, \qquad 1 \leq k \leq N_{\Delta t},$$

thus, exploiting the Lipschitz continuity of Ψ', we obtain

$$\|\tilde{\mathbf{z}}_h^k - \tilde{\mathbf{u}}_h^k\| = \|\Psi'(\mathbf{z}_n^k) - \Psi'(\mathbf{u}_n^k)\| \leq Lip(\Psi')\|\mathbf{z}_n^k - \mathbf{u}_n^k\| \leq Lip(\Psi')C\varepsilon.$$

Then, we derive C, assessing its independence from ε, under the assumption of Lipschitz continuity of the increment function Φ with respect to the second variable, of constant $Lip(\Phi)$[2], being a sufficient condition guaranteeing zero-stability of the employed method [87]. Indeed, letting $\boldsymbol{\xi}_n^k = \mathbf{z}_n^k - \mathbf{u}_n^k$, we have that

$$\boldsymbol{\xi}_n^k = \boldsymbol{\xi}_n^{k-1} + \Delta t\Big(\Phi(t_{k-1}, \mathbf{z}_n^{k-1}; \mathbf{f}_n, \Delta t) - \Phi(t_{k-1}, \mathbf{u}_n^{k-1}; \mathbf{f}_n, \Delta t) + \boldsymbol{\delta}_n^k\Big),$$

then, summing over k, with $m = 1, \ldots, N_{\Delta t}$, we obtain

$$\boldsymbol{\xi}_n^m = \boldsymbol{\xi}_n^0 + \Delta t \sum_{k=1}^{m} \boldsymbol{\delta}_n^k + \Delta t \sum_{k=1}^{m} \Big(\Phi(t_{k-1}, \mathbf{z}_n^{k-1}; \mathbf{f}_n, \Delta t) - \Phi(t_{k-1}, \mathbf{u}_n^{k-1}; \mathbf{f}_n, \Delta t)\Big).$$

Thus, thanks to the Lipschitz continuity of Φ and Ψ, it follows that

$$\begin{aligned}
\|\boldsymbol{\xi}_n^m\| &\leq \|\boldsymbol{\xi}_n^0\| + \Delta t \sum_{k=1}^{m} \|\boldsymbol{\delta}_n^k\| + \Delta t Lip(\Phi) \sum_{k=1}^{m} \|\boldsymbol{\xi}_n^{k-1}\|, && 1 \leq m \leq N_{\Delta t}, \\
&\leq Lip(\Psi)\|\boldsymbol{\delta}_h^0\| + \Delta t \sum_{k=1}^{m} \|\boldsymbol{\delta}_n^k\| + \Delta t Lip(\Phi) \sum_{k=1}^{m} \|\boldsymbol{\xi}_n^{k-1}\|, && 1 \leq m \leq N_{\Delta t}, \\
&\leq (Lip(\Psi) + m\Delta t)\varepsilon e^{m\Delta t Lip(\Phi)}, && 1 \leq m \leq N_{\Delta t}, \qquad (3.15) \\
&\leq \underbrace{(Lip(\Psi) + |T - t_0|)e^{Lip(\Phi)|T-t_0|}}_{C}\varepsilon
\end{aligned}$$

with (3.15) following from the discrete Gronwall lemma. Thus, setting

$$\tilde{C} = Lip(\Psi')(Lip(\Psi) + |T - t_0|)e^{Lip(\Phi)|T-t_0|}$$

in (3.14), we can conclude that the zero-stability property holds for the ΔLDM scheme.

Error decomposition. Since the practical implementation of LDMs is obtained by means of its discrete formulation ΔLDM, it is necessary to extend the previously derived error bounds by including the error source related to the numerical approximation employed within the latent space. In particular, by referring to (3.3), it holds that

$$\begin{aligned}
\|\mathbf{u}_h(t_k; \boldsymbol{\mu}) - \tilde{\mathbf{u}}_h^k(\boldsymbol{\mu})\| &= \|\mathbf{u}_h(t_k; \boldsymbol{\mu}) - \tilde{\mathbf{u}}_h(t_k; \boldsymbol{\mu}) + \tilde{\mathbf{u}}_h(t_k; \boldsymbol{\mu}) - \tilde{\mathbf{u}}_h^k(\boldsymbol{\mu})\| \\
&\leq \|\mathbf{u}_h(t_k; \boldsymbol{\mu}) - \tilde{\mathbf{u}}_h(t_k; \boldsymbol{\mu})\| + \|\tilde{\mathbf{u}}_h(t_k; \boldsymbol{\mu}) - \tilde{\mathbf{u}}_h^k(\boldsymbol{\mu})\| \\
&\leq C_1 + Lip(\Psi')C\Delta t^p.
\end{aligned}$$

Thus, the final form of the error decomposition, accounting for *(i)* the continuous-time approximation, namely, the upper bound terms of (3.3) (collected in C_1), and *(ii)* the numerical error source ($C_2\Delta t^p$), takes the form

$$\|\mathbf{u}_h(t_k, \boldsymbol{\mu}) - \tilde{\mathbf{u}}_h^k(\boldsymbol{\mu})\| \leq C_1 + C_2\Delta t^p, \qquad (3.16)$$

as depicted in Figure 3.

3.3 Learnable latent dynamics models

The LDP formulation described so far provides a well-structured framework that can be readily cast within a learning context, where maps equipped with learnable parameters can approximate the components involved in the LDM definition. In particular, we consider the problem of learning a LDM in a time-discrete setting, i.e. a ΔLDM. We proceed by considering *(i)* a partition of the time interval

[2]Independent of the discretization step Δt and of the nodes t_k.

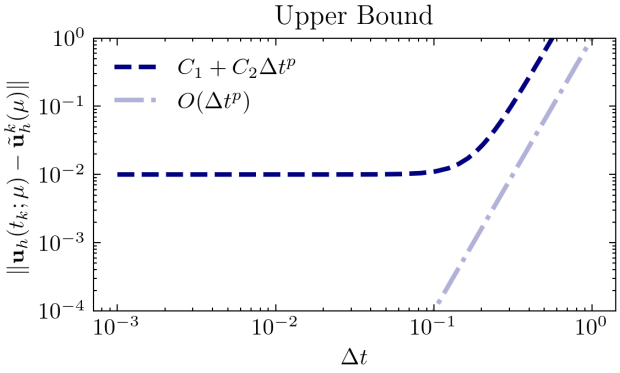

Figure 3: *Error decomposition.* Illustration of the upper bound referring to the error decomposition formula (3.16) in the discrete case, for the full-order state approximation $\tilde{\mathbf{u}}_h^k$ provided by the ΔLDM scheme.

$[t_0, T]$, with a discretization step Δt, resulting in a sequence of $N_{\Delta t} + 1$ time instances $\{t_k\}_{k=0}^{N_{\Delta t}}$, with $t_k = t_0 + k\Delta t$, and *(ii)* a proper sampling criterion over the compact set of parameters $\mathcal{P} \subset \mathbb{R}^{n_\mu}$, leading to a discrete set of N_μ sampled parameters instances $\{\boldsymbol{\mu}^j\}_{j=1}^{N_\mu}$. Then, a collection of high-fidelity FOM snapshots $\mathbf{u}_h(t_k;\boldsymbol{\mu}^j)$, sampled in correspondence of $\{t_k\}_{k=0}^{N_{\Delta t}} \times \{\boldsymbol{\mu}^j\}_{j=1}^{N_\mu}$, is considered. So, the *learnable* ΔLDM$_\theta$ approximating the evolution of a high-fidelity trajectory $\mathbf{u}_h(t;\boldsymbol{\mu})$ by means of $\hat{\mathbf{u}}_h^k(\boldsymbol{\mu})$, for a given parameter instance $\boldsymbol{\mu} \in \mathcal{P}$, reads as

$$\begin{cases} \hat{\mathbf{u}}_h^k(\boldsymbol{\mu}) = \Psi'_\theta(\hat{\mathbf{u}}_n^k(\boldsymbol{\mu})), & k = 1, \dots, N_{\Delta t}, \\ \hat{\mathbf{u}}_n^k(\boldsymbol{\mu}) = \hat{\mathbf{u}}_n^{k-1}(\boldsymbol{\mu}) + \Delta t \Phi(t_{k-1}, \hat{\mathbf{u}}_n^{k-1}(\boldsymbol{\mu}); \Delta t, \mathbf{f}_{n,\theta}, \boldsymbol{\mu}), & k = 1, \dots, N_{\Delta t}, \\ \hat{\mathbf{u}}_n^0(\boldsymbol{\mu}) = \Psi_\theta(\mathbf{u}_{0,h}(\boldsymbol{\mu})), \end{cases} \tag{3.17}$$

with $\boldsymbol{\theta} = (\boldsymbol{\theta}_\Psi, \boldsymbol{\theta}_{\Psi'}, \boldsymbol{\theta}_{\mathbf{f}_n}) \in \Theta$ denoting the vector of learnable parameters, in a suitable space Θ, associated to the triple of learnable functions $(\Psi_\theta, \Psi'_\theta, \mathbf{f}_{n,\theta})$, satisfying the following minimization problem

$$\min_{\theta \in \Theta} \mathcal{L}(\theta), \qquad \mathcal{L}(\theta) = \frac{1}{N_\mu N_{\Delta t}} \sum_{j=1}^{N_\mu} \sum_{k=1}^{N_{\Delta t}} \|\mathbf{u}_h(t_k;\boldsymbol{\mu}^j) - \hat{\mathbf{u}}_h^k(\boldsymbol{\mu}^j)\|^2. \tag{3.18}$$

Specifically, in Algorithm 1 we report a simplified version of the training procedure, highlighting the main steps to be carried out during one iteration of the ΔLDM$_\theta$ training phase, for approximating a single FOM trajectory $\mathbf{u}_h(t_k;\boldsymbol{\mu})$, sampled at temporal nodes t_k, for a fixed parameter instance $\boldsymbol{\mu} \in \mathcal{P}$. Hereafter, we explore the problem of learning a time-continuous approximation, by analyzing the approximations involved in the introduced learnable framework.

Algorithm 1 ΔLDM$_\theta$ training algorithm (single trajectory)

Input: FOM trajectory $\{\mathbf{u}_h(t_k;\boldsymbol{\mu})\}_{k=0}^{N_{\Delta t}}$, timesteps $\{t_k\}_{k=0}^{N_{\Delta t}}$, parameter instance $\boldsymbol{\mu} \in \mathcal{P}$

1: Project initial state $\hat{\mathbf{u}}_n^0(\boldsymbol{\mu}) \leftarrow \Psi_\theta(\mathbf{u}_h^0(\boldsymbol{\mu})) \in \mathbb{R}^n$
2: **for** $k = 1 : N_{\Delta t}$ **do**
3: Latent Runge-Kutta step $\hat{\mathbf{u}}_n^k \leftarrow \hat{\mathbf{u}}_n^{k-1} + \Delta t \Phi(t_{k-1}, \hat{\mathbf{u}}_n^{k-1}(\boldsymbol{\mu}); \Delta t, \mathbf{f}_{n,\theta}, \boldsymbol{\mu})$
4: Reconstruct evolved state $\hat{\mathbf{u}}_h^k(\boldsymbol{\mu}) \leftarrow \Psi'_\theta(\hat{\mathbf{u}}_n^k(\boldsymbol{\mu})) \in \mathbb{R}^{N_h}$
5: **end for**
6: Take optimization step on $\mathcal{L}(\theta)$ in (3.18).

3.3.1 Approximation results

On the basis of the ΔLDM$_\theta$ framework just introduced and the perfect embedding assumption 3.4, we aim at providing one of the main results of this work, which is contained in the Theorem 3.5, and is endowed with a constructive proof founded on the approximation results of [46]. In particular, we aim at

proving that, for each small enough Δt, the approximation error between the original FOM solution and the ROM approximation, provided by the ΔLDM_θ, is bounded. Additionally, we show that the validity of this error bound directly ensures the satisfaction of the time-continuous approximation property by the ΔLDM_θ formulation.

Theorem 3.5. *Let $\mathcal{T}_{\Delta t}$ be a general partition of $[t_0, T]$ with a discretization step Δt, resulting in a sequence of $N_{\Delta t}$ time instances $\mathcal{T}_{\Delta t} = \{t_k\}_{k=1}^{N_{\Delta t}}$. Then, under the perfect embedding assumption, for any tolerance $\varepsilon > 0$, there exist $n > 0$,*

- *a dynamics network $(t, \boldsymbol{\mu}) \mapsto \mathbf{F}_{n,\theta}(t, \boldsymbol{\mu})$, having at most $O(n\varepsilon^{-(n_\mu+1)})$ active weights,*
- *a decoder $\Psi'_{n,\theta} : \mathbb{R}^n \to \mathbb{R}^{N_h}$, having at most $O(N_h\varepsilon^{-n})$ active weights,*
- *an encoder $\Psi_{n,\theta} : \mathbb{R}^{N_h} \to \mathbb{R}^n$, having at most $O(n\varepsilon^{-N_h})$ active weights,*

and $\Delta t^* = \Delta t^*(\mathbf{F}_{n,\theta})$, such that

$$\sup_{k\in\{1,\dots,N_{\Delta t}\}} \|\mathbf{u}_h(t_k;\boldsymbol{\mu}) - \hat{\mathbf{u}}_h^k(\boldsymbol{\mu})\| \le \varepsilon, \quad \forall \Delta t \le \Delta t^*, \quad \forall \boldsymbol{\mu} \in \mathcal{P}, \tag{3.19}$$

where

$$\hat{\mathbf{u}}_h^k(\boldsymbol{\mu}) := \Psi'_{n,\theta}(\mathbf{u}_{n,\theta}^k(\boldsymbol{\mu})) = \Psi'_{n,\theta}\Big(\Psi_{n,\theta}(\mathbf{u}_h(t_0;\boldsymbol{\mu})) + \Delta t \sum_{j=0}^{k} w_j \mathbf{F}_{n,\theta}(s_j;\boldsymbol{\mu})\Big), \qquad \forall k = 1,\dots,N_{\Delta t}, \tag{3.20}$$

with $(w_j, s_j)_{j=1}^{N_{\Delta t}}$ pairs of quadrature weights and nodes.

Proof. We let $\varepsilon > 0$. Moreover, without loss of generality, here we assume $t_0 = 0$. The proof is divided into 5 parts.

(i) *Definition of the exact latent dynamics*
By employing the definition of LDM and the perfect embedding assumption, we can state that there exists $n^* \ll N_h$ such that, for any $n \ge n^*$ there exists $\Psi'_n, \Psi_n, \mathbf{f}_n$, such that

$$\begin{cases} \mathbf{u}_h(t;\boldsymbol{\mu}) = \tilde{\mathbf{u}}_h(t;\boldsymbol{\mu}) = \Psi'_n(\mathbf{u}_n(t;\boldsymbol{\mu})), & t \in (0,T], \\ \dot{\mathbf{u}}_n(t;\boldsymbol{\mu}) = \mathbf{f}_n(t, \mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu}), & t \in (0,T], \\ \mathbf{u}_n(0;\boldsymbol{\mu}) = \Psi_n(\mathbf{u}_h(0;\boldsymbol{\mu})), \end{cases}$$

for any $\boldsymbol{\mu} \in \mathcal{P}$. Thus, calling $\mathbf{F}_n(t;\boldsymbol{\mu}) = \mathbf{f}_n(t, \mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})$, for any $t \in (0,T]$, we define explicitly

$$\mathbf{u}_n(t;\boldsymbol{\mu}) = \Psi_n(\mathbf{u}_h(0;\boldsymbol{\mu})) + \int_0^t \mathbf{F}_n(s;\boldsymbol{\mu})ds.$$

(ii) *Definition of the neural network architectures*
Since $\mathbf{F}_n(t;\boldsymbol{\mu})$ is Lipschitz-continuous as function of $(t,\boldsymbol{\mu})$, thanks to Gühring-Raslan Theorem [46], there exists a neural network $(t,\boldsymbol{\mu}) \mapsto \mathbf{F}_{n,\theta}(t;\boldsymbol{\mu})$ consisting of at most $O(n\varepsilon^{-(n_\mu+1)})$ active weights such that

$$\|\mathbf{F}_n - \mathbf{F}_{n,\theta}\|_{W^{0,\infty}([0,T]\times\mathcal{P};\mathbb{R}^n)} \le (8T)^{-1}\varepsilon.$$

Moreover, we highlight that the manifold $\mathcal{S}_0 = \{\mathbf{u}_h(0;\boldsymbol{\mu}) : \boldsymbol{\mu} \in \mathcal{P}\}$ is such that $diam(\mathcal{S}_0)$ is bounded since $\mathcal{P}$ is compact. Thus, owing to $\Psi_n \in C^1(\mathbb{R}^{N_h};\mathbb{R}^n)$, it is possible to construct a neural network architecture $\Psi_{n,\theta} : \mathcal{S}_0 \to \mathbb{R}^n$ having $O(n\varepsilon^{-N_h})$ active weights, such that

$$\|\Psi_n(\mathbf{u}_h(0;\boldsymbol{\mu})) - \Psi_{n,\theta}(\mathbf{u}_h(0;\boldsymbol{\mu}))\|_{W^{0,\infty}(\mathcal{P};\mathbb{R}^n)} \le 8^{-1}\varepsilon.$$

Then, it is possible to prove that $\mathcal{S}_n = \{\mathbf{u}_n(t;\boldsymbol{\mu}) : t \in [0,T],\ \boldsymbol{\mu} \in \mathcal{P}\}$ has bounded diameter. Indeed, we emphasize that $[0,T] \times \mathcal{P}$ is compact and by definition of $\mathbf{u}_n$, since Ψ_n and $\mathbf{F}_n$ are bounded with respect to t and Lipschitz-continuous with respect to $\boldsymbol{\mu}$, we have, $\forall t_1, t_2 \in [0,T]$ and $\forall \boldsymbol{\mu}_1, \boldsymbol{\mu}_2 \in \mathcal{P} \subset \mathbb{R}^{n_\mu}$,

$$\begin{aligned}
&\|\mathbf{u}_n(t_1;\boldsymbol{\mu}_1) - \mathbf{u}_n(t_2;\boldsymbol{\mu}_2)\| \leq \|\mathbf{u}_n(t_1;\boldsymbol{\mu}_1) - \mathbf{u}_n(t_2;\boldsymbol{\mu}_1)\| + \|\mathbf{u}_n(t_2;\boldsymbol{\mu}_1) - \mathbf{u}_n(t_2;\boldsymbol{\mu}_2)\| \\
&\leq \|\mathbf{F}_n(t;\boldsymbol{\mu})\|_{W^{0,\infty}([0,T]\times\mathcal{P};\mathbb{R}^n)}|t_1 - t_2| + Lip(\Psi_n)\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\| + \int_0^{t_2} \|\mathbf{F}_n(s;\boldsymbol{\mu}_1) - \mathbf{F}_n(s;\boldsymbol{\mu}_2)\|ds \\
&\leq \|\mathbf{F}_n(t;\boldsymbol{\mu})\|_{W^{0,\infty}([0,T]\times\mathcal{P};\mathbb{R}^n)}|t_1 - t_2| + Lip(\Psi_n)\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\| + TLip(\mathbf{F}_n)\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|.
\end{aligned}$$

Thus, applying Gühring-Raslan Theorem, it is possible to construct a neural network $\Psi'_{n,\theta} : \mathcal{S}_n \to \mathbb{R}^{N_h}$ having at most $O(N_h\varepsilon^{-n})$ active weights such that

$$\|\Psi'_n(\mathbf{u}_n(t;\boldsymbol{\mu})) - \Psi'_{n,\theta}(\mathbf{u}_n(t;\boldsymbol{\mu}))\|_{W^{0,\infty}([0,T]\times\mathcal{P};\mathbb{R}^{N_h})} \leq 2^{-1}\varepsilon.$$

(iii) *Approximation of the latent dynamics at a time-continuous level*
Hence, we are now able to define the explicit form of the LDM_θ at a continuous level, namely

$$\mathbf{u}_{n,\theta}(t;\boldsymbol{\mu}) = \Psi_{n,\theta}(\mathbf{u}_h(0;\boldsymbol{\mu})) + \int_0^t \mathbf{F}_{n,\theta}(s;\boldsymbol{\mu})ds.$$

We remark that we let $\mathbf{u}_{n,\theta}(t;\mu)$ (and $\mathbf{u}_n(t;\mu)$) absorbs $Lip(\Psi'_{n,\theta})$ so that, without loss of generality, we can assume $Lip(\Psi'_{n,\theta}) = 1$. Then, it is straightforward to verify that

$$\begin{aligned}
E_{CONT} :&= \|\mathbf{u}_n - \mathbf{u}_{n,\theta}\|_{W^{0,\infty}([0,T]\times\mathcal{P};\mathbb{R}^n)} \\
&= \sup_{t\in[0,T]} \left\| \Psi_n(\mathbf{u}_h(0;\boldsymbol{\mu})) + \int_0^t \mathbf{F}_n(s;\boldsymbol{\mu})ds - \Psi_{n,\theta}(\mathbf{u}_h(0;\boldsymbol{\mu})) + \int_0^t \mathbf{F}_{n,\theta}(s;\boldsymbol{\mu})ds \right\|_{W^{0,\infty}(\mathcal{P};\mathbb{R}^n)} \\
&\leq \|\Psi_n(\mathbf{u}_h(0;\boldsymbol{\mu})) - \Psi_{n,\theta}(\mathbf{u}_h(0;\boldsymbol{\mu}))\|_{W^{0,\infty}(\mathcal{P};\mathbb{R}^n)} + \\
&\qquad + \sup_{t\in[0,T]} \int_0^t \|\mathbf{F}_n(s;\boldsymbol{\mu}) - \mathbf{F}_{n,\theta}(s;\boldsymbol{\mu})\|_{W^{0,\infty}(\mathcal{P};\mathbb{R}^n)}ds \\
&\leq 8^{-1}\varepsilon + T\|\mathbf{F}_n - \mathbf{F}_{n,\theta}\|_{W^{0,\infty}([0,T]\times\mathcal{P};\mathbb{R}^n)} \\
&\leq 8^{-1}\varepsilon + T(8T)^{-1}\varepsilon = 4^{-1}\varepsilon.
\end{aligned}$$

(iv) *Approximation of the latent dynamics at a time-discrete level*
Let $\Delta t > 0$, then $\mathcal{T}_{\Delta t}$ is a uniform partition of $[0,T]$. We set $N_{\Delta t} = \text{card}(\mathcal{T}_{\Delta t})$ and we define, for any $k = 1, \ldots, N_{\Delta t}$,

$$\mathbf{u}^k_{n,\theta}(\boldsymbol{\mu}) := \Psi_{n,\theta}(\mathbf{u}_h(0;\boldsymbol{\mu})) + \Delta t \sum_{j=0}^{k} w_j \mathbf{F}_{n,\theta}(s_j;\boldsymbol{\mu}),$$

where $(w_j, s_j)_{j=1}^{N_{\Delta t}}$ are the quadrature weights and nodes that implicitly depend on Δt. We define the quadrature error as

$$E_{QUAD}(\Delta t) := \sup_{k\in\{1,\ldots,N_{\Delta t}\}} \left\| \int_0^{t_k} \mathbf{F}_{n,\theta}(s;\boldsymbol{\mu})ds - \Delta t \sum_{j=0}^{k} w_j \mathbf{F}_{n,\theta}(s_j;\boldsymbol{\mu}) \right\|_{W^{0,\infty}(\mathcal{P};\mathbb{R}^n)}.$$

If the scheme is convergent, then there exists a monotonically increasing function $\Delta t \mapsto h(\Delta t)$ such that $h(\Delta t) \to 0$, as $\Delta t \to 0$, and a constant $C > 0$ for which $E_{QUAD}(\Delta t) \leq Ch(\Delta t)$. Observe that $\Delta t \mapsto h(\Delta t)$ implicitly depends on $\mathbf{F}_{n,\theta}$. Then, there exists $\Delta t^* = \Delta t^*(\varepsilon; \mathbf{F}_{n,\theta}) > 0$ such that for any $\Delta t \leq \Delta t^*$ it holds that

$$E_{QUAD}(\Delta t) \leq Ch(\Delta t) \leq Ch(\Delta t^*) \leq 4^{-1}\varepsilon.$$

Thus, hereon, we let $\Delta t \leq \Delta t^*$. Being $\mathcal{T}_{\Delta t} \subset [0,T]$ and by employing the triangular inequality we

obtain

$$
\begin{aligned}
E_{DISCR} :&= \sup_{k\in\{1,\dots,N_{\Delta t}\}} \|\mathbf{u}_n(t_k) - \mathbf{u}^k_{n,\theta}\|_{W^{0,\infty}(\mathcal{P};\mathbb{R}^n)} \\
&\leq \sup_{k\in\{1,\dots,N_{\Delta t}\}} \left[\|\mathbf{u}_n(t_k) - \mathbf{u}_{n,\theta}(t_k)\|_{W^{0,\infty}(\mathcal{P};\mathbb{R}^n)} + \|\mathbf{u}_{n,\theta}(t_k) - \mathbf{u}^k_{n,\theta}\|_{W^{0,\infty}(\mathcal{P};\mathbb{R}^n)}\right] \\
&\leq \|\mathbf{u}_n - \mathbf{u}_{n,\theta}\|_{W^{0,\infty}([0,T]\times\mathcal{P};\mathbb{R}^n)} + \sup_{k\in\{1,\dots,N_{\Delta t}\}} \|\mathbf{u}_{n,\theta}(t_k) - \mathbf{u}^k_{n,\theta}\|_{W^{0,\infty}(\mathcal{P};\mathbb{R}^n)} \\
&= E_{CONT} + \sup_{k\in\{1,\dots,N_{\Delta t}\}} \|\mathbf{u}_{n,\theta}(t_k) - \mathbf{u}^k_{n,\theta}\|_{W^{0,\infty}(\mathcal{P};\mathbb{R}^n)} \\
&= E_{CONT} + \sup_{k\in\{1,\dots,N_{\Delta t}\}} \left\| \int_0^{t_k} \mathbf{F}_{n,\theta}(s;\boldsymbol{\mu})ds - \Delta t \sum_{j=0}^{k} w_j \mathbf{F}_{n,\theta}(s_j;\boldsymbol{\mu})\right\|_{W^{0,\infty}(\mathcal{P};\mathbb{R}^n)} \\
&= E_{CONT} + E_{QUAD}(\Delta t) \leq 2^{-1}\varepsilon.
\end{aligned}
$$

(v) *Final error bound*
Defining

$$
\hat{\mathbf{u}}^k_h(\boldsymbol{\mu}) := \Psi'_{n,\theta}(\mathbf{u}^k_{n,\theta}(\boldsymbol{\mu})) = \Psi'_{n,\theta}\left(\Psi_{n,\theta}(\mathbf{u}_h(0;\boldsymbol{\mu})) + \Delta t \sum_{j=0}^{k} w_j \mathbf{F}_{n,\theta}(s_j;\boldsymbol{\mu})\right),
$$

through the triangular inequality, we obtain

$$
\begin{aligned}
\sup_{k\in\{1,\dots,N_{\Delta t}\}} &\|\mathbf{u}_h(t_k;\boldsymbol{\mu}) - \hat{\mathbf{u}}^k_h(\boldsymbol{\mu})\|_{W^{0,\infty}(\mathcal{P};\mathbb{R}^{N_h})} \leq \\
&\leq \sup_{k\in\{1,\dots,N_{\Delta t}\}} \Big[\|\Psi'_n(\mathbf{u}_n(t_k;\boldsymbol{\mu})) - \Psi'_{n,\theta}(\mathbf{u}_n(t_k;\boldsymbol{\mu}))\|_{W^{0,\infty}(\mathcal{P};\mathbb{R}^{N_h})} \\
&\quad + \|\Psi'_{n,\theta}(\mathbf{u}_n(t_k;\boldsymbol{\mu})) - \Psi'_{n,\theta}(\mathbf{u}^k_{n,\theta}(\boldsymbol{\mu}))\|_{W^{0,\infty}(\mathcal{P};\mathbb{R}^{N_h})}\Big] \\
&\leq E_{DEC} + E_{LAT}.
\end{aligned}
$$

Then, we can bound E_{DEC} with

$$
\begin{aligned}
E_{DEC} :&= \sup_{k\in\{1,\dots,N_{\Delta t}\}} \|\Psi'_n(\mathbf{u}_n(t_k;\boldsymbol{\mu})) - \Psi'_{n,\theta}(\mathbf{u}_n(t_k;\boldsymbol{\mu}))\|_{W^{0,\infty}(\mathcal{P};\mathbb{R}^{N_h})} \\
&\leq \|\Psi'_n(\mathbf{u}_n(t;\boldsymbol{\mu})) - \Psi'_{n,\theta}(\mathbf{u}_n(t;\boldsymbol{\mu}))\|_{W^{0,\infty}([0,T]\times\mathcal{P};\mathbb{R}^{N_h})} \leq 2^{-1}\varepsilon.
\end{aligned}
$$

Moreover, thanks to the $\Psi'_{n,\theta}$ being $1-Lip$, we obtain

$$
\begin{aligned}
E_{LAT} :&= \sup_{k\in\{1,\dots,N_{\Delta t}\}} \|\Psi'_{n,\theta}(\mathbf{u}_n(t_k;\boldsymbol{\mu})) - \Psi'_{n,\theta}(\mathbf{u}^k_{n,\theta}(\boldsymbol{\mu}))\|_{W^{0,\infty}(\mathcal{P};\mathbb{R}^{N_h})} \\
&\leq \sup_{k\in\{1,\dots,N_{\Delta t}\}} \|\mathbf{u}_n(t_k;\boldsymbol{\mu}) - \mathbf{u}^k_{n,\theta}(\boldsymbol{\mu})\|_{W^{0,\infty}(\mathcal{P};\mathbb{R}^n)} = E_{DISCR} \leq 2^{-1}\varepsilon.
\end{aligned}
$$

Finally, we can conclude the proof, obtaining that

$$
\sup_{k\in\{1,\dots,N_{\Delta t}\}} \|\mathbf{u}_h(t_k;\boldsymbol{\mu}) - \hat{\mathbf{u}}^k_h(\boldsymbol{\mu})\|_{W^{0,\infty}(\mathcal{P};\mathbb{R}^{N_h})} = E_{DEC} + E_{LAT} \leq \varepsilon.
$$

□

Remark 3.6. We remark that Eq.(3.20) can be traced back to the original ΔLDM_θ formulation (3.17). Moreover, we notice that the encoder architecture may suffer from the curse of dimensionality, as the number of active weights may scale as $O(\varepsilon^{-N_h})$, which becomes prohibitive as N_h increases. However, the encoder can be replaced by a feedforward neural network $\boldsymbol{\mu} \mapsto \mathbf{g}(\boldsymbol{\mu}) \approx \Psi(\mathbf{u}_h(0;\boldsymbol{\mu}))$ in a DL-ROM fashion [10, 33], thus requiring at most $O(n\varepsilon^{-n_\mu})$ weights.

Time-continuity in a learnable context. Having introduced the ΔLDM_θ, we aim at characterizing, within such learnable setting, the notion of *time-continuous approximation* defined for a generic ROM in Section 2. In particular, the dynamical nature of the ΔLDM_θ, characterized by a *continuous-time* inductive bias represented by the (discretized) latent IVP, allows us to query the learned solution with arbitrary precision by using suitably fine temporal discretizations that, in principle, could be different from the one employed during the learning procedure. In such a context, the time-continuous approximation property follows from the approximation result given in Theorem 3.5, as highlighted by the following remark.

Remark 3.7. By setting Δt^* in Definition 2.1 equal to $\Delta t^*(\mathbf{F}_{n,\theta})$ introduced in the proof of Theorem 3.5, it is straightforward to revisit Eq. (3.19) as a time-continuity property, in agreement with Eq. (2.5).

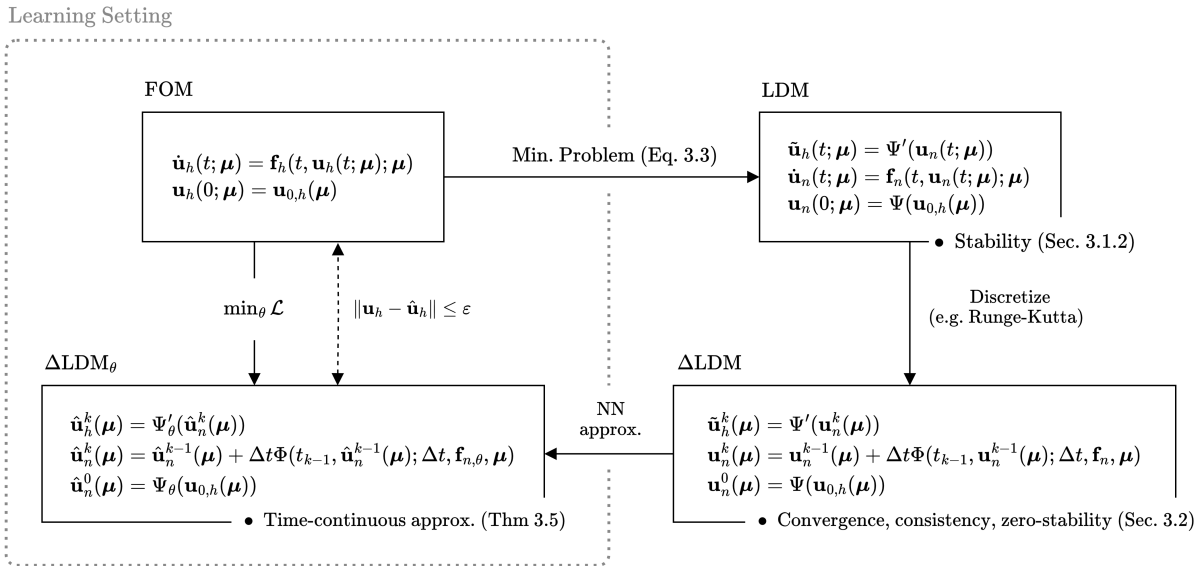

Figure 4: *Framework overview.* The diagram summarizes the proposed framework, considering the learnable setting (ΔLDM_θ), the time-continuous (LDM) and time-discrete (ΔLDM) approximations, together with the respective main properties.

4 Deep learning-based LDMs

In the following section, the architectural choices regarding the design space of the proposed framework, in a DL context, are described. As introduced in Section 3.3, the aim of a *learnable* LDM is to model the evolution of a parameterized FOM state in a data-driven manner, by means of learnable maps $\Psi_\theta, \Psi'_\theta$ tackling the dimensionality reduction task, and a learnable latent dynamics $\mathbf{f}_{n,\theta}$ modeling the latent state evolution. In particular, the identification of a ROM state $\mathbf{u}_n(t;\boldsymbol{\mu})$, associated to a full-order parameterized time-dependent state $\mathbf{u}_h(t;\boldsymbol{\mu})$, and the modeling of its time-evolution, is a task of critical importance in the context of PDEs surrogate modeling.

To this end, a commonly adopted data-driven approach consists of employing: *(i)* nonlinear dimensionality reduction strategies, such as *autoencoders* (AEs) [53], to learn a meaningful latent representation lying on a low-dimensional manifold [59, 69], and *(ii)* a proper time-stepping scheme, in order to model the time-evolution of the latent state. Multiple data-driven approaches, combining the concept of dimensionality reduction and latent time-evolution, have been proposed, mainly by relying on autoregressive or recurrent strategies [21, 34, 39, 43] in time-discrete settings.

A more recent approach consists in relying on continuous-time modeling strategies, via *neural ordinary differential equations* (NODEs) [16], to model the evolution of the latent state [17, 25, 29, 68, 70, 71]. In particular, the family of models that combines AEs and NODEs, specialized in the context of multi-query reduced order modeling by [70], aligns with our theoretical LDM formulation. Specifically, within the introduced ΔLDM_θ learnable framework, an AE neural network identifies the composition $\Psi'_\theta \circ \Psi_\theta$, while

a parameterized NODE represents the latent ODE $\dot{\mathbf{u}}_n(t;\boldsymbol{\mu}) = \mathbf{f}_{n,\theta}(t, \mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})$, within a DL context. Thus, building up on such framework, we proceed to describe the architectural choices that underlie the construction of DL-based LDMs.
In this respect, we remark that the proposed framework does not put constraints on the underlying neural networks' design. On the other hand, it is clear that the obtained theoretical results are related to the approximation error and do not take into account the optimization error. However, in practice, reducing the contribution of the optimization error is crucial to ensure an acceptable accuracy in the inference phase. To do that, ultimately aiming at enhancing the convergence of the training procedure, we now introduce suitable architectural choices and we propose an adequate training strategy.

4.1 Spatially-coherent autoencoding

Recent approaches address the dimensionality reduction of a full-order state $\mathbf{u}_h(t;\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ in a data-driven manner, by relying on AEs within an unsupervised learning setting. Specifically, AEs based on convolutional architectures are often adopted, due to their ability to leverage spatial information and their sparse nature, which allows them to handle high-dimensional data effectively while maintaining a reasonable parameter count [33, 35, 69, 91].
In particular, we proceed by considering the full-order solution $\mathbf{u}_h(t;\boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ of (2.1), obtained by means of a spatial-discretization of the associated PDE, via traditional high-fidelity techniques, such as the Galerkin-finite element method. These methods entail the introduction of a computational grid, denoted as Ω_h, serving as a discretization of the bounded domain $\Omega \subset \mathbb{R}^d$, with h the spatial discretization parameter, representing the maximum diameter of the grid elements. The choice of h influences the granularity of the discretization, that is, smaller values result in finer resolutions, leading to higher FOM dimensions N_h, and thus to increased computational costs. In traditional high-fidelity techniques, the solution is represented by a vector $\mathbf{u}_h \in \mathbb{R}^{N_h}$ corresponding to the DoFs of the problem. However, this representation could lack *spatial coherence*, especially in the context of unstructured grids. In such scenarios, neighboring entries in the vector may correspond to spatial points that are not necessarily adjacent in both the physical and computational domains. This lack of spatial coherence prevents convolutional architectures from effectively leveraging local spatial information. Additionally, using reshape operations, whether at the input level to adjust tensor shapes or due to the inclusion of dense layers requiring flattening operations, further restricts the exploitation of spatial information.
Thus, we aim to address these issues, by adopting a dimensionality reduction strategy capable of retaining spatial coherence, by exploiting spatial information through a fully-convolutional architecture, not relying on reshaping operations and dense layers. To introduce our framework, we consider the reference computational grid Ω_h, possibly unstructured, employed for solving the FOM. To enable spatial-coherence, we introduce a structured grid $\Omega_h^\natural$, obtained via a uniform discretization[3] of Ω. Rather than relying on reshaping operations, to suitably adapt the shape of the FOM state to be processed via convolutional layers – which would not lead to a spatially coherent representation – the state $\mathbf{u}_h \in \mathbb{R}^{N_h}$ computed on Ω_h is interpolated onto the uniform grid $\Omega_h^\natural$, by means of a suitable interpolation operator $I_\natural$. As a consequence, we obtain a state vector $I_\natural \mathbf{u}_h \in \mathbb{R}^{N_h}$ which exhibits a spatially-coherent ordering, and is thus prepared to be inputted into the convolutional architecture, thanks to the adopted uniform discretization[4].
In the following, the internal structure of the convolutional AE is described. Specifically, the architecture relies on a composition of convolutional layers and interpolation operations. Both the encoder Ψ_θ and decoder Ψ'_θ are structured into L down- and up-sampling stages, respectively indicated as $s_l : \mathbb{R}^{2^{-(l-1)}N_h} \to \mathbb{R}^{2^{-l}N_h}$ and $s'_l : \mathbb{R}^{2^{-l}N_h} \to \mathbb{R}^{2^{-(l-1)}N_h}$, with $l = 1, \dots, L$. Such stages are then composed, resulting in $\Psi_\theta = s_L \circ s_{L-1} \circ \cdots \circ s_1$ and $\Psi'_\theta = s'_1 \circ s'_2 \circ \cdots \circ s'_L$, as shown in the following

$$\mathbf{u}_h \in \mathbb{R}^{N_h} \quad \underset{s'_1}{\overset{s_1}{\rightleftarrows}} \quad \mathbb{R}^{2^{-1}N_h} \quad \underset{s'_2}{\overset{s_2}{\rightleftarrows}} \quad \cdots \quad \underset{s'_L}{\overset{s_L}{\rightleftarrows}} \quad \mathbb{R}^{2^{-L}N_h} \ni \mathbf{u}_n$$

[3] The novel discretization $\Omega_h^\natural$ uses as discretization parameter a value as close as possible to the original one h, resulting in the same number of DoFs N_h, thereby we avoid introducing additional notation.

[4] Since the method will always rely on $I_\natural$ to process the FOM state, we will refer to the spatially-coherent FOM state $I_\natural \mathbf{u}_h$ simply as $\mathbf{u}_h$.

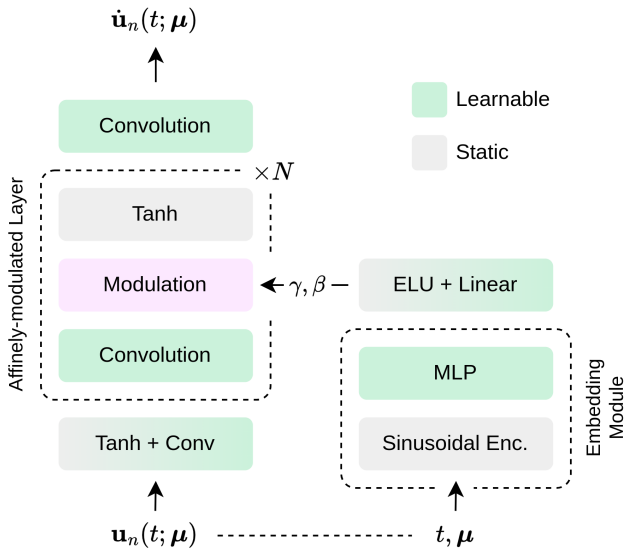

Figure 5: *Affinely-parameterized latent dynamics architecture.* Detailed structure of the proposed convolutional parameterized latent dynamics. The two main components – the embedding module, which processes the time-parameters inputs, and the stack of affinely-modulated convolutional layers – are highlighted.

with the first row referring to the encoder, while the second one refers to the decoder; here, we denote by $n = 2^{-L}N_h$ the reduced state dimension. Both l-th level stages, s_l and s'_l, rely on the same internal structure, reading respectively as

$$s_l = I^{2^l h}_{2^{l-1}h} \circ R_l, \qquad s'_l = I^{2^{l-1}h}_{2^l h} \circ R'_l, \tag{4.1}$$

where:

(i) R_l and R'_l refer to a preactivation *residual* convolutional block [49], with learnable convolutional kernels $\kappa_{in}, \kappa_{out}$, and suitable nonlinearity σ, both reading as

$$\mathbf{h}_{l-1} \mapsto \mathbf{h}_{l-1} + \kappa_{out} * \sigma(\kappa_{in} * \sigma(\mathbf{h}_{l-1})). \tag{4.2}$$

with $\mathbf{h}_{l-1}$ a hidden state, output of the $(l-1)$-th level;

(ii) $I^{2^l h}_{2^{l-1}h} : \mathbb{R}^{2^{-(l-1)}N_h} \to \mathbb{R}^{2^{-l}N_h}$ and $I^{2^{l-1}h}_{2^l h} : \mathbb{R}^{2^{-l}N_h} \to \mathbb{R}^{2^{-(l-1)}N_h}$ are the interpolation operators. The former, in s_l, performs a down-sampling operation by halving the spatial dimension, resulting in a coarsening operator. On the other hand, within s'_l, the operation is reversed, to perform up-sampling.

Concerning the implementation details, we employ convolutional layers with a kernel size of 3. Both the encoder and the decoder are equipped with a linear convolutional layer, as a final layer, after the last interpolation. We adopt an ELU nonlinearity [19], being continuously differentiable with bounded first derivative, thus satisfying the Lipschitz requirement on Ψ_θ and Ψ'_θ. We emphasize that our approach is motivated by the utilization of regular domains. For the case of domains characterized by irregularities, or more complex geometries, approaches such as graph neural networks (GNNs) or mesh-informed strategies may be more suitable options [31, 32, 85]; however, these aspects are beyond the focus of this work, and can represent further extensions of the proposed framework.

4.2 Affinely-parameterized latent dynamics

The concept of *neural ordinary differential equation* (NODE) [16, 58, 75] represents an implicitly-defined class of models which – unlike traditional modeling approaches directly establishing an input-output

relation $\mathbf{x} \mapsto \mathbf{y}$ as $\mathbf{y} = \mathbf{f}_\theta(\mathbf{x})$ – adopts the following ODE-IVP formulation

$$\dot{\mathbf{y}}(t) = \mathbf{f}_\theta(t, \mathbf{y}(t)), \qquad \mathbf{y}(0) = \mathbf{x}.$$

Here, $\mathbf{f}_\theta$ is now a neural network parameterizing the vector field by which the input is continuously evolved into the output over a prescribed time interval $t \in [0, T]$ (see Appendix A). The inherently continuous-time nature of this family of models makes them well-suited for modeling time-dependent data [92]. Specifically, their formulation has led to a wide adoption of such architectures in the field of data-driven reduced order modeling, to model the reduced dynamics $\mathbf{f}_{n,\theta}$ in combination with AEs [17, 25, 70, 71]. Recently, [70] extended the NODE formulation to incorporate parametric information represented by $\boldsymbol{\mu} \in \mathcal{P}$. This enhancement improves the expressivity of the neural network, resulting in the concept of parameterized neural ODEs (PNODEs). In contrast to the prevailing approaches of using dense neural networks to parameterize the PNODE dynamics, we explore the adoption of a convolutional architecture. This choice is motivated by the intrinsic sparsity of convolutional layers, resulting in a smaller network size footprint. Moreover, such choice allows us to leverage the spatial information encoded by the convolutional AE, possibly enabling a spatial-coherence between the encoded and the high-dimensional states, thus enhancing interpretability. In the following, the architectural choices in the design of an *affinely-parameterized latent dynamics* $\mathbf{f}_{n,\theta}$, are described. Specifically, we delineate the embedding strategy employed for the temporal and parametric information $(t, \boldsymbol{\mu})$, by means of *affine modulation* techniques, departing from traditional concatenation or hypernetwork-based approaches.

Time-parameters embedding. A modeling challenge is represented by the effective embedding of the time $t \in \mathbb{R}_+$ and parameters $\boldsymbol{\mu} \in \mathcal{P}$ into the reduced representation, leading to a parameterized latent state $\mathbf{u}_n(t; \boldsymbol{\mu})$. Such problem is naively tackled by directly providing $(t, \boldsymbol{\mu})$ as inputs to the neural network architecture, after performing a rescaling operation to normalize their ranges. In recent NODE-based ROM approaches, the main strategies for injecting additional information into the dynamics is represented by either concatenation or by relying on hypernetworks [70, 107]. Our approach departs from standard parameters embedding techniques. In particular, for embedding temporal and parametric information, we draw inspiration from recent advances in embedding strategies widely adopted in state-of-the-art DL architectures, from diffusion models to transformers [98]. Rather than using a basic rescaling of scalar features on t and the components μ_j of $\boldsymbol{\mu}$, we propose to map them to a higher-dimensional vector space $\mathbb{R}^k$ via a *sinusoidal encoding*, where $k \in \mathbb{N}$ (even) represents the number of frequency components. In particular, the employed mappings are $t \mapsto \tilde{\mathbf{t}} \in \mathbb{R}^k$ for the time feature, and $\boldsymbol{\mu} \mapsto \tilde{\boldsymbol{\mu}} \in \mathbb{R}^{kn_\mu}$ for the parameters vector, since the embedding is applied component-wise to each scalar parameter $\mu_j,\ j = 1, \dots, n_\mu$. Moreover, sinusoidal encoding allows mapping scalar features to vectors with entries bounded in the range $[-1, 1]$, thereby avoiding the need for separate rescaling operations. The specific form of the employed encoding reads as

$$t \mapsto \tilde{\mathbf{t}}(t) = \left[\sin(\omega_1 t) \quad \sin(\omega_2 t) \quad \cdots \quad \sin(\omega_{\frac{k}{2}} t) \quad \cos(\omega_1 t) \quad \cos(\omega_2 t) \quad \cdots \quad \cos(\omega_{\frac{k}{2}} t)\right], \tag{4.3}$$

$$\omega_j = \frac{1}{T_{\max}^{j-1}}, \qquad j = 1, ..., \frac{k}{2},$$

where $T_{\max}$ is a large enough period, allowing the scalar input t to be represented by a k-dimensional signal $\tilde{\mathbf{t}}$ of period $2\pi T_{\max}^{\frac{k}{2}-1}$, thus exceeding the typical time-scales of the modeled phenomena ($\gg T_{\text{train}}, T_{\text{test}}$). The mapping (4.3) can be extended in a straightforward way to the scalar components μ_j of the parameters vector $\boldsymbol{\mu} \in \mathbb{R}^{n_\mu}$ to perform the encoding of each component, resulting in $\tilde{\boldsymbol{\mu}}$. Subsequently, the encoded features $(\tilde{\mathbf{t}}, \tilde{\boldsymbol{\mu}})$ are concatenated and processed by a multi-layer perceptron (MLP) with ELU nonlinearity, resulting in a combined embedding $\boldsymbol{\xi} \in \mathbb{R}^{d_e}$, containing both the temporal and parametric information. Similar embedding approaches, based on sinusoidal encoding, have been adopted in autoregressive architectures employed in the context of DL-based surrogate modeling, for embedding either the temporal or parametric information [45, 72]. Finally, the embedded information $\boldsymbol{\xi}$ is integrated into the convolutional dynamics $\mathbf{f}_{n,\theta}$ via *affine modulation* strategies, as detailed in the following.

Parameterization via affine modulation. Effectively including additional information, alongside the primary input, to guide and control learning, is a task of central importance in DL. In particular,

in the context of multi-query DL-based surrogate modeling, this translates into the task of including the information related to $\boldsymbol{\mu} \in \mathcal{P}$ to construct the surrogate model. Moreover, embedding the time-parameters information is of paramount importance in the specific case of learning a parameterized dynamics $\mathbf{f}_{n,\theta}(t, \ \cdot \ ; \boldsymbol{\mu})$, taking into account both the temporal and parametric information $(t, \boldsymbol{\mu}) \in \mathbb{R}_+ \times \mathcal{P}$. A possible approach to tackle this problem consists in using hypernetworks, where an auxiliary network takes the parametric information as input and outputs the weights of the main network, represented in our case by the latent dynamics $\mathbf{f}_{n,\theta}$. While effective, this approach often results in models of considerable size, due to the necessity to infer the main network's entire set of parameters, thus hindering scalability and possibly leading to memory constraints and computational overhead. To cope with this issue, [107] proposes a factorization-based approach, applied to the dense layers of the ODE-Net modeling the latent dynamics. Specifically, a hypernetwork is employed to infer only the diagonal entries $\Sigma(\boldsymbol{\mu})_{ij}$ of a SVD-like decomposition of the weights matrices $W_\theta(\boldsymbol{\mu}) = U\Sigma(\boldsymbol{\mu})V^T$ of the latent dynamics dense layers, thus reducing the cost associated to the hypernetwork. In contrast, mainly motivated by our fully-convolutional architecture, we adopt an *affine modulation* approach, drawing inspiration from recent advancements in conditional image generation, where class or textual information influences the generative process. In particular, in conditional normalization-based methods [5, 56], a function of the conditioning input $\boldsymbol{\xi}$ is learned to output the normalization layers scaling $\boldsymbol{\gamma}(\boldsymbol{\xi})$ and shifting $\boldsymbol{\beta}(\boldsymbol{\xi})$ parameters, performing an affine transformation of the following form

$$\tilde{\mathbf{x}} = \boldsymbol{\gamma}(\boldsymbol{\xi}) \odot \mathbf{x} \ \oplus \ \boldsymbol{\beta}(\boldsymbol{\xi}), \tag{4.4}$$

with $\mathbf{x}$ an arbitrary (possibly hidden) feature of the network being conditioned, to which a prior normalization technique has been applied. In particular, such conditioning technique has found broad application in diffusion models, to incorporate the timestep and class embedding into the residual blocks of the UNet employed in the diffusion process [24, 79]. A more general framework, complementing the previous ones, is represented by [84], which employs *feature-wise* affine transformations to modulate the network's intermediate features, based on additional input sources. More broadly, as pointed out in [84], these methods of injecting conditional signals via affine feature modulation can be viewed as instances of hypernetwork architectures [47], where a subsidiary network outputs the parameters of the main network, based on the conditioning input. However, affine modulation schemes are much more efficient since the number of parameters required for the affine transformation is significantly smaller than the main network's total parameters count.
Building up on affine modulation techniques, we define a convolutional affinely-parameterized NODE, to include both time t and parameters $\boldsymbol{\mu}$ into the latent dynamics $\mathbf{f}_{n,\theta}$, by means of the embedding $\boldsymbol{\xi}(t, \boldsymbol{\mu})$. Thus, relying on the conditioned affine transformation (4.4) to include the embedded parametric information $\boldsymbol{\xi}(t, \boldsymbol{\mu})$, we can define an affinely-modulated convolutional layer, representing the core of the convolutional *affinely-parameterized latent dynamics* $\mathbf{f}_{n,\theta}$ illustrated in Figure 5, reading as

$$\mathbf{h}_{i-1} \mapsto \boldsymbol{\gamma}_i(\boldsymbol{\xi}) \odot (\kappa_i * \mathbf{h}_{i-1}) \ \oplus \ \boldsymbol{\beta}_i(\boldsymbol{\xi}), \tag{4.5}$$

where κ_i represent the learnable kernel of the i-th convolutional layer, with n_c channels, and $\mathbf{h}_{i-1}$ the modulated hidden state. The affine-parameterization is performed via the scaling and shifting parameters $\boldsymbol{\gamma}_i(\boldsymbol{\xi}), \boldsymbol{\beta}_i(\boldsymbol{\xi}) \in \mathbb{R}^{n_c}$, which modulate the i-th layer preactivation along the channel dimension, via channel-wise multiplication $\odot$ and sum $\oplus$, respectively. The dependence of $\boldsymbol{\gamma}_i(\boldsymbol{\xi}), \boldsymbol{\beta}_i(\boldsymbol{\xi})$ on the time and parameter instances $(t, \boldsymbol{\mu})$ is implicit, by means of $\boldsymbol{\xi}(t, \boldsymbol{\mu})$, produced by the embedding module previously introduced. In particular, the mapping $\boldsymbol{\xi} \mapsto (\boldsymbol{\gamma}_i, \boldsymbol{\beta}_i)$ is modeled by a linear layer preceded by a nonlinearity σ, reading as $W\sigma(\ \cdot \ ) \ + \mathbf{b} : \mathbb{R}^{d_e} \to \mathbb{R}^{2n_c}$. The choice of performing the affine transformation along the channels dimension makes the $(t, \boldsymbol{\mu})$-parameterization agnostic to the spatial dimension of the latent state, decoupling the size of the embedding network from the latent state spatial dimension, thus making the technique scalable and computationally efficient.
Regarding the implementation details, convolutional layers with a kernel size of 3 are employed. Regarding the employed activation functions, an ELU nonlinearity is adopted within the embedding module, while the modulated convolutional layers employ tanh, both being Lipschitz continuous. Finally, in Algorithm 2, we outline the internal structure of the proposed *affinely-parameterized latent dynamics* $\mathbf{f}_{n,\theta}$, summarizing the components described above, in the case of two modulated convolutional layers of kernels κ_1, κ_2.

Algorithm 2 Affinely-parameterized latent dynamics $\mathbf{f}_{n,\theta}$

Input: Latent state $\mathbf{u}_n(t;\boldsymbol{\mu}) \in \mathbb{R}^n$, time instance $t \in \mathbb{R}_+$, parameter instance $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^{n_\mu}$
Output: Latent state time-derivative $\dot{\mathbf{u}}_n(t;\boldsymbol{\mu}) \in \mathbb{R}^n$

1: Encode time instance $\tilde{\mathbf{t}} \leftarrow \text{SinusoidalEncoding}(t, k, T_{\max}) \in \mathbb{R}^k$
2: Encode parameter instance $\tilde{\boldsymbol{\mu}} \leftarrow \text{SinusoidalEncoding}(\boldsymbol{\mu}, k, T_{\max}) \in \mathbb{R}^{kn_\mu}$
3: Time-parameter embedding $\boldsymbol{\xi} = \text{MLP}(\tilde{\mathbf{t}}, \tilde{\boldsymbol{\mu}}) \in \mathbb{R}^{d_e}$
4: Affine modulation parameters $\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2 \leftarrow W\sigma(\boldsymbol{\xi}) + \mathbf{b} \in \mathbb{R}^{4n_c}$
5: Input convolution $\mathbf{h} = \tanh(\kappa_{\text{in}} * \mathbf{u}_n)$
6: Convolve and modulate $\mathbf{h} = \tanh(\boldsymbol{\gamma}_1 \odot (\kappa_1 * \mathbf{h}) \ \oplus \ \boldsymbol{\beta}_1)$
7: Convolve and modulate $\mathbf{h} = \tanh(\boldsymbol{\gamma}_2 \odot (\kappa_2 * \mathbf{h}) \ \oplus \ \boldsymbol{\beta}_2)$
8: Output convolution $\dot{\mathbf{u}}_n(t;\boldsymbol{\mu}) = \kappa_{\text{out}} * \mathbf{h}$

Remark 4.1. Adopting a bounded activation function within the parameterized dynamics $\mathbf{f}_{n,\theta}$, as tanh, enables a straightforward way to estimate $\|\mathbf{f}_{n,\theta}\|_\infty$. Indeed, considering a 1-dimensional convolutional kernel κ_{out} of size $2k+1$ with c input channels and 1 output channel, and given tanh bounded range $(-1, 1)$, it follows that

$$\|\mathbf{f}_{n,\theta}\|_\infty \leq \sum_{i=1}^{c} \sum_{i=-k}^{k} |\kappa_{\text{out}}^{ij}|.$$

4.3 Training and testing schemes

In the following, the training and testing procedures for the DL-based LDM are described. We proceed by denoting with $\mathbf{S}_h \in \mathbb{R}^{N_h \times N_s}$ and $\mathbf{M} \in \mathbb{R}^{(n_\mu+1)\times N_s}$ the matrices collecting the high-fidelity solutions $\mathbf{u}_h(t_i;\boldsymbol{\mu}^j)$, and the associated time-parameter instances $(t_i, \boldsymbol{\mu}^j)$, respectively. We indicate by $N_s = N_t N_\mu$ the total number of available snapshots, with N_t the length of the collected trajectories, and N_μ the total number of trajectories computed for different instances of the parameters $\boldsymbol{\mu}^j$, resulting in

$$\mathbf{S}_h = \left[\mathbf{u}_h(t_0;\boldsymbol{\mu}^1) \mid \cdots \mid \mathbf{u}_h(t_{N_t-1};\boldsymbol{\mu}^1) \mid \cdots \mid \mathbf{u}_h(t_0;\boldsymbol{\mu}^{N_\mu}) \mid \cdots \mid \mathbf{u}_h(t_{N_t-1};\boldsymbol{\mu}^{N_\mu}) \right],$$

$$\mathbf{M} = \left[\begin{matrix} \boldsymbol{\mu}^1 \\ t_0 \end{matrix} \middle| \cdots \middle| \begin{matrix} \boldsymbol{\mu}^1 \\ t_{N_t-1} \end{matrix} \middle| \cdots \middle| \begin{matrix} \boldsymbol{\mu}^{N_\mu} \\ t_0 \end{matrix} \middle| \cdots \middle| \begin{matrix} \boldsymbol{\mu}^{N_\mu} \\ t_{N_t-1} \end{matrix} \right].$$

Before the start of the training routine, the training and validation snapshot matrices $\mathbf{S}_h^{train}, \mathbf{S}_h^{val}$, obtained by means of a splitting along the temporal and parameters dimensions, accordingly to splitting ratios $\alpha, \beta \in (0, 1)$, undergo a normalization step to rescale the features in the [-1,1] range, reading as

$$\mathbf{S}_h^* \mapsto 2 \cdot \frac{\mathbf{S}_h^* \ - \ \min(\mathbf{S}_h^{train})}{\max(\mathbf{S}_h^{train}) - \min(\mathbf{S}_h^{train})} - 1, \tag{4.6}$$

with $\mathbf{S}_h^*$ indicating either $\mathbf{S}_h^{train}$ or $\mathbf{S}_h^{val}$. When dealing with a FOM state composed of multiple scalar components, transformation (4.6) is applied separately, rescaling each component. As follows, Algorithm 3 and 4 provide a detailed description of the training and testing schemes, respectively. Specifically, the outlined training procedure builds up on the minimization problem (3.18) and Algorithm 1. In particular, given a collection of snapshots $\mathbf{S}_h^*$, either belonging to the training or validation set, we denote by $\mathbf{S}_{0,h}^*$ the collection of the initial values $\{\mathbf{u}_h(t_0;\boldsymbol{\mu}^j)\}_{j=1}^{N_\mu}$, obtained by extracting the slice referring to the initial time instance t_0. Then, $\mathbf{S}_{0,n}^*$ refers to the collection of *latent* initial values, obtained by applying the encoder function Ψ_θ to $\mathbf{S}_{0,h}^*$. Similarly, $\mathbf{S}_n^*$ refers to the evolution of the latent states, obtained by integrating the latent dynamics $\mathbf{f}_{n,\theta}$ accordingly to a prescribed RK scheme. Then $\hat{\mathbf{S}}_h^*$ denotes the LDM approximation of the FOM state evolution, output of the decoder function Ψ'_θ, applied to the previously computed latent representations $\mathbf{S}_n^*$. Given the approximation $\hat{\mathbf{S}}_h^*$, the inverse transformation of (4.6) is applied in order to rescale the model output to the original range. The optimization procedure involved in the training, outlined in Algorithm 3 (8-15), relies on Adam algorithm [60] to perform the model

parameters update, by employing a suitable learning rate η and weight decay λ.

Algorithm 3 ΔLDM_θ training algorithm (w/ sub-trajectories)

Input: Snapshot matrix $\mathbf{S}_h \in \mathbb{R}^{N_h \times N_s}$, parameter matrix $\mathbf{M} \in \mathbb{R}^{(n_\mu+1)\times N_s}$, time-parameter training-validation split ratios $\alpha, \beta \in (0,1)$, learning rate η, weight decay λ, batch size N_b, sub-trajectories length $2 \leq \ell \leq \min\{N_t^{train}, N_t^{val}\}$, maximum number of epochs N_{epochs}, early-stopping criterion, Runge-Kutta scheme RK

Output: Optimal model parameters $\boldsymbol{\theta}^* = (\boldsymbol{\theta}_\Psi^*, \boldsymbol{\theta}_{\Psi'}^*, \boldsymbol{\theta}_{\mathbf{f}_n}^*)$

Split data $\mathbf{S}_h = [\mathbf{S}_h^{train}, \mathbf{S}_h^{val}]$, $\mathbf{M} = [\mathbf{M}^{train}, \mathbf{M}^{val}]$ with $N_t^{train} = \alpha N_t$, $N_t^{val} = (1-\alpha)N_t$, and $N_\mu^{train} = \beta N_\mu$, $N_\mu^{val} = (1-\beta)N_\mu$

Normalize $\mathbf{S}_h$ accordingly to (4.6)

Assemble *train* sub-traj. $\tilde{\mathbf{S}}_h^{train} \in \mathbb{R}^{N_h \times \ell \times N_\mu^{train}(N_t^{train}-\ell+1)}$, $\tilde{\mathbf{M}}^{train} \in \mathbb{R}^{(n_\mu+1)\times \ell \times N_\mu^{train}(N_t^{train}-\ell+1)}$ with $N_{batches}^{train} = N_\mu^{train}(N_t^{train}-\ell+1)/N_b$

Assemble *val* sub-traj. $\tilde{\mathbf{S}}_h^{val} \in \mathbb{R}^{N_h \times \ell \times N_\mu^{val}(N_t^{val}-\ell+1)}$, $\tilde{\mathbf{M}}^{val} \in \mathbb{R}^{(n_\mu+1)\times \ell \times N_\mu^{val}(N_t^{val}-\ell+1)}$ with $N_{batches}^{val} = N_\mu^{val}(N_t^{val}-\ell+1)/N_b$

i = 0

while ($\neg$early-stopping $\wedge$ $i \leq N_{epochs}$) **do**

  for $k = 1 : N_{batches}^{train}$ **do**

    Get k-th mini-batch $(\mathbf{S}_h^{batch}, \mathbf{M}^{batch}) \subseteq (\tilde{\mathbf{S}}_h^{train}, \tilde{\mathbf{M}}^{train})$

    Extract initial values $\mathbf{S}_{0,h}^{batch}$ from $\mathbf{S}_h^{batch}$

    Extract time and parameters instances $(\mathbf{t}^{batch}, \boldsymbol{\mu}^{batch}) \leftarrow \mathbf{M}^{batch}$

    Project initial values $\mathbf{S}_{0,n}^{batch} = \Psi_\theta(\mathbf{S}_{0,h}^{batch}; \boldsymbol{\theta}_\Psi)$

    Latent ODE integration $\mathbf{S}_n^{batch} = \text{RK}(\mathbf{f}_{n,\theta}, \mathbf{S}_{0,n}^{batch}, \mathbf{t}^{batch}, \boldsymbol{\mu}^{batch}; \boldsymbol{\theta}_{\mathbf{f}_n})$

    Reconstruct trajectories $\hat{\mathbf{S}}_h^{batch} = \Psi'_\theta(\mathbf{S}_n^{batch}; \boldsymbol{\theta}_{\Psi'})$

    Accumulate loss $\mathcal{L}(\boldsymbol{\theta})$ on $(\mathbf{S}_h^{batch}, \mathbf{M}^{batch})$, compute $\nabla_\theta \mathcal{L}$

    Update model parameters $\boldsymbol{\theta} \leftarrow \text{Adam}(\eta, \lambda, \nabla_\theta \mathcal{L}, \boldsymbol{\theta})$

  end for

  Repeat (7-13) on $(\mathbf{S}_h^{val}, \mathbf{M}^{val})$

  Accumulate loss $\mathcal{L}(\boldsymbol{\theta})$ on $(\mathbf{S}_h^{val}, \mathbf{M}^{val})$

  i++

end while

Remark 4.2 (*Temporal regularization*). To reduce the dependence on the sub-trajectories' length ℓ, a temporal regularization approach [41] may be adopted, involving randomly sampled integration interval lengths. It can be implemented in Algorithm 3 by fixing a maximum sub-trajectory length $\ell_{max} > 2$, used to construct the training matrices $\mathbf{S}_h^{train}, \mathbf{M}^{train}$. At each iteration, a trajectory length is then sampled $\ell \sim \mathcal{U}[2, \ell_{max}]$, allowing the selection of a batch with sub-trajectory length ℓ as $\mathbf{S}_{h,0:\ell}^{batch}$, $\mathbf{M}_{0:\ell}^{batch}$.

Algorithm 4 ΔLDM_θ testing algorithm

Input: Initial snapshot matrix $\mathbf{S}_{0,h}^{test} \in \mathbb{R}^{N_h \times N_\mu}$, parameter matrix $\mathbf{M}^{test} \in \mathbb{R}^{(n_\mu+1)\times N_s}$, optimal model parameters $\boldsymbol{\theta}^* = (\boldsymbol{\theta}_\Psi^*, \boldsymbol{\theta}_{\Psi'}^*, \boldsymbol{\theta}_{\mathbf{f}_n}^*)$, Runge-Kutta scheme RK, perturbation $\boldsymbol{\delta}_h \in \mathbb{R}^{N_h}$

Output: Approximation $\hat{S}_h^{test} \in \mathbb{R}^{N_h \times N_s}$

Normalize $\mathbf{S}_{0,h}^{test}$ accordingly to (4.6)

if $\boldsymbol{\delta}_h \neq \mathbf{0}$ **then**

  Perturb initial values $\mathbf{S}_{0,h}^{test} \leftarrow \mathbf{S}_{0,h}^{test} + \boldsymbol{\delta}_h$

end if

Extract timesteps and parameters $(\mathbf{t}^{test}, \boldsymbol{\mu}^{test}) \leftarrow \mathbf{M}^{test}$

$\mathbf{S}_{0,n} = \Psi_\theta(\mathbf{S}_{0,h}^{test}; \boldsymbol{\theta}_\Psi^*)$

$\mathbf{S}_n = \text{RK}(\mathbf{f}_{n,\theta}, \mathbf{S}_{0,n}, \mathbf{t}^{test}, \boldsymbol{\mu}^{test}; \boldsymbol{\theta}_{\mathbf{f}_n}^*)$

$\hat{\mathbf{S}}_h^{test} = \Psi'_\theta(\mathbf{S}_n; \boldsymbol{\theta}_{\Psi'}^*)$

Inversely normalize $\hat{\mathbf{S}}_h^{test}$ accordingly to $\neg$(4.6)

Additionally, the testing Algorithm 4 includes the option to perturb the initial datum (line 3), to test LDMs' zero-stability via a suitable perturbation $\boldsymbol{\delta}_h$.

5 Numerical results

This section is concerned with the evaluation of the performance of the proposed framework in the context of reduced order modeling of parameterized dynamical systems, i.e., FOMs of the form (2.1), arising from the semi-discretization of parameterized nonlinear time-dependent PDEs. The goal of the following numerical tests is to assess LDMs' capabilities in *(i)* accurately handling time evolution, *(ii)* satisfying the time-continuous approximation property, *(iii)* ensuring zero-stability in a learnable context, and *(iv)* effectively capturing the parametric dependence in a multi-query context.

5.1 Problems and experimental setup

The considered problems on which empirical tests are performed are *(i)* a one-dimensional Burger's equation, and *(ii)* a two-dimensional advection-diffusion-reaction (ADR) equation. The specific setup of each problem is described below.

1D Burgers' equation. As a first problem, the one-dimensional Burgers' equation is considered, reading as

$$\begin{cases} u_t - \nu u_{xx} + u u_x = 0 & (x,t) \in \Omega \times (0,T], \\ u_x(x,t) = 0 & (x,t) \in \partial\Omega \times (0,T], \\ u(x,0) = e^{-x^2} & x \in \Omega. \end{cases} \tag{5.1}$$

Here $\nu \in \mathbb{R}$ denotes the viscosity coefficient, varying in the interval $\nu \in \mathcal{P} = [5 \cdot 10^{-3}, 1]$, having in this case $\dim \mathcal{P} = n_\mu = 1$. Regarding the parameter space splitting and discretization, we consider $\mathcal{P} = [5 \cdot 10^{-3}, 5 \cdot 10^{-2}] \cup (5 \cdot 10^{-2}, 1]$, where the former interval is used for training and interpolation testing purposes, while the latter is reserved for extrapolation testing. In particular, the first interval is discretized into $N_\mu^{train} = 100$ equally spaced instances, forming the training set $\mathcal{P}_{train}$, where 20% of these instances are kept for validation. For testing purposes with respect to the parametric dependence, two discrete sets are constructed. $\mathcal{P}_{interp}$ refers to the interpolation parameter instances, obtained by taking the midpoints of the training instances in $\mathcal{P}_{train}$, resulting in $N_\mu^{interp} = 99$ data points. Considering the extrapolation testing instances, $\mathcal{P}_{extrap}$ is constructed by considering $N_\mu^{extrap} = 50$ equally spaced points over $(5 \cdot 10^{-2}, 1]$. The FOM is built by employing linear ($\mathbb{P}_1$) finite elements, with a spatial grid characterized by $N_h = 1024$ DoFs over $\Omega = (-10, 10)$. The system has been solved in time via implicit differentiation of order 1, with $N_t = 1000$ time steps over the interval $[0, T]$ with $T = 30$, resulting in a time step $\Delta t_{FOM} = 0.03$. A temporal splitting of the form $[0, T_1] \cup (T_1, T_2] \cup (T_2, T]$ is adopted, where the first two intervals refer to the training and validation partitions, while the latter refers to the time-extrapolation interval. Specifically, $T_1 = 12$, $T_2 = 15$ are chosen as the endpoints of the training and validation intervals, respectively.

2D Advection-diffusion-reaction equation. As a second test case, we consider a time-dependent parameterized advection-reaction-diffusion problem on a two-dimensional domain $\Omega = (0,1)^2$, defined as follows

$$\begin{cases} u_t - \nabla \cdot (\mu_1 \nabla u) + \mathbf{b}(t) \cdot \nabla u + cu = f(\mathbf{x}; \mu_2, \mu_3) & (\mathbf{x}, t) \in \Omega \times (0,T], \\ \mu_1 \nabla u \cdot \mathbf{n} = 0 & (\mathbf{x}, t) \in \partial\Omega \times (0,T], \\ u(\mathbf{x}, 0) = 0 & \mathbf{x} \in \Omega, \end{cases} \tag{5.2}$$

with

$$f(\mathbf{x}; \mu_2, \mu_3) = 10 \cdot \exp(-((x-\mu_2)^2 + (y-\mu_3)^2)/0.07^2), \qquad \mathbf{b}(t) = [\cos(t),\ \sin(t)]^T, \qquad c = 1.$$

In this case $n_\mu = 3$, so that the parameters left varying are $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3)$, where the former refers to the diffusive effect, while the latter two set the position of the forcing term, thus acting on a geometrical aspect of the problem. In particular, the nonlinear dependence of the solution on μ_2 and μ_3 categorizes

the problem as nonaffinely parameterized, posing a challenge for traditional projection-based techniques [23]. The adopted parameters space is $\mathcal{P} = [1 \cdot 10^{-2}, 6 \cdot 10^{-2}] \times [0.3, 0.7]^2$. Specifically, considering $[2 \cdot 10^{-2}, 5 \cdot 10^{-2}] \times [0.4, 0.6]^2 \subset \mathcal{P}$, 10 uniformly spaced instances are taken over each interval, resulting in $N_\mu^{train} = 1000$ training parameters instances collected into $\mathcal{P}_{train}$, with 20% of these instances reserved for validation. As in the previous test case, the midpoints of the training set instances are collected into $\mathcal{P}_{interp}$, leading to $N_\mu^{interp} = 729$ interpolation testing instances. The model's extrapolation capabilities are tested on $N_\mu^{extrap} = 200$ instances uniformly sampled from $\mathcal{P} \setminus [2 \cdot 10^{-2}, 5 \cdot 10^{-2}] \times [0.4, 0.6]^2$, forming $\mathcal{P}_{extrap}$. The FOM is solved through a linear ($\mathbb{P}_1$) finite elements discretization in space, and considering 32 nodes on each side of Ω, resulting in $N_h = 1024$ DoFs. The system has been solved in time via implicit differentiation of order 1, with $N_t = 1000$ time steps over the interval $[0, T]$ with $T = 10\pi$, thus with a time step $\Delta t_{FOM} \simeq 0.03$. A temporal splitting of the form $[0, T_1] \cup (T_1, T_2] \cup (T_2, T]$ is again adopted, choosing $T_1 = \frac{2}{5}T,\ T_2 = \frac{1}{2}T$ as the endpoints of the training and validation intervals, respectively.

Experimental setup. The architecture of the models employed in the proposed numerical experiments follows the structure given in Section 4, characterized by a convolutional autoencoder and a convolutional parameterized neural ODE, equipped with the proposed affine modulation parameterization strategy. The training procedure outlined in Algorithm 1 is followed, adopting sub-trajectories of maximum sequence length $\ell_{max} = 40$, together with the *temporal regularization* method outlined in Remark 4.2. Adam optimizer [60] is employed with a decaying learning rate schedule starting from $\eta = 5 \cdot 10^{-4}$, and weight decay $\lambda = 10^{-5}$. The models' implementation has been performed using the PyTorch framework. Specifically, the ODE integration numerical routines have been implemented from scratch, rather than relying on external libraries, in order to deal with parameter-dependent latent dynamics, and leverage just-in-time (JIT) compilation for improved training and inference performance. For the purpose of evaluating LDMs' performance, the following error indicators are introduced:

(i) a scalar indicator $\epsilon_{rel}(\mathbf{u}_h, \hat{\mathbf{u}}_h) \in \mathbb{R}$ representing the relative error averaged over N_μ parameters' instances and N_t temporal steps[5] ($N_s = N_t N_\mu$), reading as

$$\epsilon_{rel}(\mathbf{u}_h, \hat{\mathbf{u}}_h) = \frac{1}{N_s} \sum_{j=1}^{N_\mu} \sum_{i=0}^{N_t - 1} \frac{\|\mathbf{u}_h(t_i; \boldsymbol{\mu}^j) - \hat{\mathbf{u}}_h(t_i; \boldsymbol{\mu}^j)\|}{\|\mathbf{u}_h(t_i; \boldsymbol{\mu}^j)\|}, \tag{5.3}$$

measuring the average performances over the time-parameter space;

(ii) a parameter-dependent scalar indicator $\epsilon_{rel}(\boldsymbol{\mu}; \mathbf{u}_h, \hat{\mathbf{u}}_h) \in \mathbb{R}$, which quantifies the relative error averaged over N_t temporal steps, for a given parameter instance $\boldsymbol{\mu}$, reading as

$$\epsilon_{rel}(\boldsymbol{\mu}; \mathbf{u}_h, \hat{\mathbf{u}}_h) = \frac{1}{N_t} \sum_{i=0}^{N_t - 1} \frac{\|\mathbf{u}_h(t_i; \boldsymbol{\mu}) - \hat{\mathbf{u}}_h(t_i; \boldsymbol{\mu})\|}{\|\mathbf{u}_h(t_i; \boldsymbol{\mu})\|}, \tag{5.4}$$

used to assess the accuracy over the parameter space.

(iii) a time-dependent scalar indicator $\epsilon_{rel}(t; \mathbf{u}_h, \hat{\mathbf{u}}_h, \boldsymbol{\mu}) \in \mathbb{R}$, representing the pointwise-in-time relative error for a fixed instance of the testing parameter $\boldsymbol{\mu}$, defined as

$$\epsilon_{rel}(t; \mathbf{u}_h, \hat{\mathbf{u}}_h, \boldsymbol{\mu}) = \frac{\|\mathbf{u}_h(t; \boldsymbol{\mu}) - \hat{\mathbf{u}}_h(t; \boldsymbol{\mu})\|}{\|\mathbf{u}_h(t; \boldsymbol{\mu})\|}, \tag{5.5}$$

(iv) a time-dependent vector indicator $\boldsymbol{\epsilon}_{rel}(t; \mathbf{u}_h, \hat{\mathbf{u}}_h, \boldsymbol{\mu}) \in \mathbb{R}^{N_h}$ providing the pointwise-in-time relative error for a fixed instance of the testing parameter $\boldsymbol{\mu}$, defined as

$$\boldsymbol{\epsilon}_{rel}(t; \mathbf{u}_h, \hat{\mathbf{u}}_h, \boldsymbol{\mu}) = \frac{|\mathbf{u}_h(t; \boldsymbol{\mu}) - \hat{\mathbf{u}}_h(t; \boldsymbol{\mu})|}{\|\mathbf{u}_h(t; \boldsymbol{\mu})\|}. \tag{5.6}$$

[5]Here, N_t may be replaced by an arbitrary sub-trajectory length $\ell \leq N_t$, such as in the case of short-term predictions.

To maintain consistency with the previously introduced relative error measures, the error indicator used to assess the time-continuous approximation property is the supremum of (5.5) over the employed time-grid $\{t_k\}_{k=0}^{N_t-1}$, namely

$$\epsilon_{rel}^{sup}(\mathbf{u}_h, \hat{\mathbf{u}}_h, \boldsymbol{\mu}) = \sup_{k \in \{0,\dots,N_t-1\}} \epsilon_{rel}(t_k; \mathbf{u}_h, \hat{\mathbf{u}}_h, \boldsymbol{\mu}). \tag{5.7}$$

5.2 Computational experiments

The following experiments aim to assess the properties and generalization capabilities of the proposed framework, with a focus on four key modeling aspects: *(i)* temporal evolution, *(ii)* time-continuity, *(iii)* zero-stability and *(iv)* parameter dependence. Although these aspects are addressed separately in the following subsections, they are closely connected throughout the different experiments. Indeed, being in a multi-query context, the performances with respect to varying parameter instances are always assessed, and, exploiting the time-continuous nature of the framework, finer testing temporal discretizations than the one employed at training-time are employed.
Regarding the models' architectural details, a latent dimension of $n = 16$ is used throughout the experiments, resulting in a dimensionality reduction by a factor of 64, being $N_h = 1024$ in both benchmarks problems. Second- and fourth-order Ralston's Runge-Kutta (RK) schemes [90] are adopted for the numerical solution of the parameterized latent NODE. Coarser training temporal discretizations, compared to the ones used for solving the FOM, are employed. In particular, denoting the training time step by Δt^*, for the first benchmark problem $\Delta t^* = 2\Delta t_{FOM}$, while for the second one $\Delta t^* = 5\Delta t_{FOM}$. Moreover, the LDMs used in the following studies are characterized by a parameter count of at most $\sim$200K, significantly smaller than that of many overparameterized DL-based ROMs.

5.2.1 Temporal evolution

Being primarily a temporal modeling scheme based on latent dynamics learning, the proposed framework is tested to assess the temporal evolution of both the high-dimensional approximation provided by the LDM, and the underlying evolution of the latent state. Specifically, two types of tests are performed: the first evaluates the *short-term* predictive capabilities of LDMs, while the second assesses their the *long-term* predictive performance. We remark that both tests are conducted on the FOM temporal discretizations, thus being $\Delta t_{test} = \Delta t^*/2$ in the Burgers' test case, and $\Delta t_{test} = \Delta t^*/5$ in the ADR test case. In the following, RK2 is employed for integrating the latent dynamics, with parameters encoding of size $k = 16$ and $k = 8$ employed in the Burgers' and ADR test cases, respectively.

		$0 < t_0 < T_1$			$T_1 < t_0 < T_2$			$T_2 < t_0 < T$		
	$\ell_{test} =$	20	40	80	20	40	80	20	40	80
Burgers	$\mathcal{P}_{train}$	$1.23 \cdot 10^{-3}$	$1.14 \cdot 10^{-3}$	$1.08 \cdot 10^{-3}$	$5.28 \cdot 10^{-3}$	$5.21 \cdot 10^{-3}$	$5.28 \cdot 10^{-3}$	$1.67 \cdot 10^{-2}$	$1.65 \cdot 10^{-2}$	$1.75 \cdot 10^{-2}$
	$\mathcal{P}_{interp}$	$1.24 \cdot 10^{-3}$	$1.10 \cdot 10^{-3}$	$1.06 \cdot 10^{-3}$	$5.11 \cdot 10^{-3}$	$4.94 \cdot 10^{-3}$	$5.08 \cdot 10^{-3}$	$1.60 \cdot 10^{-2}$	$1.60 \cdot 10^{-2}$	$1.72 \cdot 10^{-2}$
	$\mathcal{P}_{extrap}$	$5.32 \cdot 10^{-3}$	$6.77 \cdot 10^{-3}$	$9.41 \cdot 10^{-3}$	$1.26 \cdot 10^{-2}$	$1.33 \cdot 10^{-2}$	$1.53 \cdot 10^{-2}$	$1.84 \cdot 10^{-2}$	$2.02 \cdot 10^{-2}$	$2.40 \cdot 10^{-2}$
ADR	$\mathcal{P}_{train}$	$5.75 \cdot 10^{-3}$	$5.56 \cdot 10^{-3}$	$5.34 \cdot 10^{-3}$	$9.23 \cdot 10^{-3}$	$1.16 \cdot 10^{-2}$	$1.41 \cdot 10^{-2}$	$1.44 \cdot 10^{-2}$	$2.35 \cdot 10^{-2}$	$3.19 \cdot 10^{-2}$
	$\mathcal{P}_{interp}$	$5.35 \cdot 10^{-3}$	$5.38 \cdot 10^{-3}$	$5.40 \cdot 10^{-3}$	$8.82 \cdot 10^{-3}$	$1.14 \cdot 10^{-2}$	$1.38 \cdot 10^{-2}$	$1.50 \cdot 10^{-2}$	$2.38 \cdot 10^{-2}$	$3.25 \cdot 10^{-2}$
	$\mathcal{P}_{extrap}$	$6.20 \cdot 10^{-2}$	$5.98 \cdot 10^{-2}$	$6.28 \cdot 10^{-2}$	$5.99 \cdot 10^{-2}$	$6.36 \cdot 10^{-2}$	$7.03 \cdot 10^{-2}$	$6.55 \cdot 10^{-2}$	$6.99 \cdot 10^{-3}$	$8.01 \cdot 10^{-2}$

Table 1: *Short-term predictions.* Relative error $\epsilon_{rel}(\mathbf{u}_h, \hat{\mathbf{u}}_h)$ for short-term predictions on sub-trajectories of test length ℓ_{test} being $\frac{1}{2}\times, 1\times, 2\times$ the maximum training sub-trajectory length $\ell_{max} = 40$. The tests are performed by sampling sub-trajectories from the train $[0, T_1]$, validation $(T_1, T_2]$ and extrapolation $(T_2, T]$ temporal intervals. Similarly, the rows correspond to the train ($\mathcal{P}_{train}$), interpolation ($\mathcal{P}_{interp}$) and extrapolation ($\mathcal{P}_{extrap}$) parameter instances.

To test short-term predictive capabilities, sub-trajectories of comparable length to the ones employed at training time are sampled from the FOM trajectories. In particular, the employed testing lengths are $\ell = 20, 40, 80$, being $\frac{1}{2}\times, 1\times, 2\times$ the maximum training sub-trajectory length $\ell_{max} = 40$ used for

LDMs' training in both Burgers' and ADR test cases. The testing sub-trajectories are extracted from the different partitions of the temporal interval $[0, T]$, namely the training split $[0, T_1]$, the validation one $(T_1, T_2]$, and the extrapolation interval $(T_2, T]$. In such setting, the extracted FOM sub-trajectory is denoted as $\{\mathbf{u}_h(t_k; \boldsymbol{\mu})\}_{k=0}^{\ell-1}$, with $\{t_k\}_{k=0}^{\ell-1}$ being a collection of time-instances belonging to one of the temporal splits. Leveraging LDM's IVP-structure, $\mathbf{u}_h(t_0; \boldsymbol{\mu})$ is provided as input to the model, along with the temporal steps $\{t_k\}_{k=0}^{\ell-1}$ and the parameters $\boldsymbol{\mu}$, to produce the approximation $\{\hat{\mathbf{u}}_h(t_k; \boldsymbol{\mu})\}_{k=0}^{\ell-1}$. The sub-trajectories are sampled for different parameter instances $\boldsymbol{\mu}$, belonging to either the training set $\mathcal{P}_{train}$, interpolation set $\mathcal{P}_{interp}$ or extrapolation set $\mathcal{P}_{extrap}$. The results, expressed in terms of the relative error indicator $\epsilon_{rel}(\mathbf{u}_h, \hat{\mathbf{u}}_h)$ (Eq. (5.3)), are collected in Table 1. In particular, it can be observed that the LDM is able to provide accurate approximations in both the Burgers' equation and ADR equation test cases, effectively generalizing in time ($t > T_2$), and when performing extrapolation with respect to the parameters ($\boldsymbol{\mu} \in \mathcal{P}_{extrap}$), with errors on the order of 10^{-2}, at most. In particular, the same error levels are maintained across different sub-trajectories lengths, highlighting LDMs' capability to handle short-term predictions of comparable length to the training sub-trajectories.

						CPU Time (s)		
		$0 < t < T_1$	$T_1 < t < T_2$	$T_2 < t < T$	$[0, T]$	FOM	LDM	Speed-up
Burgers	$\mathcal{P}_{train}$	$3.32 \cdot 10^{-3}$	$7.66 \cdot 10^{-3}$	$2.95 \cdot 10^{-2}$	$1.68 \cdot 10^{-2}$	4.90	$9.13 \cdot 10^{-2}$	
	$\mathcal{P}_{interp}$	$3.17 \cdot 10^{-3}$	$7.16 \cdot 10^{-3}$	$2.89 \cdot 10^{-2}$	$1.65 \cdot 10^{-2}$	4.65	$7.84 \cdot 10^{-2}$	$\sim 40\times$
	$\mathcal{P}_{extrap}$	$2.67 \cdot 10^{-2}$	$4.01 \cdot 10^{-2}$	$6.14 \cdot 10^{-2}$	$4.55 \cdot 10^{-2}$	4.41	$9.94 \cdot 10^{-2}$	
ADR	$\mathcal{P}_{train}$	$6.21 \cdot 10^{-3}$	$1.31 \cdot 10^{-2}$	$4.75 \cdot 10^{-2}$	$2.73 \cdot 10^{-2}$	9.25	$3.74 \cdot 10^{-2}$	
	$\mathcal{P}_{interp}$	$5.94 \cdot 10^{-3}$	$1.27 \cdot 10^{-2}$	$4.67 \cdot 10^{-2}$	$2.67 \cdot 10^{-2}$	9.31	$3.67 \cdot 10^{-2}$	$\sim 200\times$
	$\mathcal{P}_{extrap}$	$7.73 \cdot 10^{-2}$	$8.74 \cdot 10^{-2}$	$1.16 \cdot 10^{-1}$	$9.81 \cdot 10^{-2}$	9.84	$3.68 \cdot 10^{-2}$	

Table 2: *Long-term predictions and CPU times.* Relative error $\epsilon_{rel}(\mathbf{u}_h, \hat{\mathbf{u}}_h)$ for long-term predictions, obtained by providing $\mathbf{u}_{0,h}$ at $t_0 = 0$, and integrating the LDM's latent dynamics over the entire temporal interval $[0, T]$. The first three columns show the error indicator values over the train, validation and extrapolation temporal splits, respectively, while the fourth column reports the relative error over the full interval. On the right, a comparison between FOM's and LDM's computational times is provided.

Long-term predictive capabilities are tested by considering the whole FOM trajectories on the complete interval $[0, T]$, resulting in the discretized sequence $\{\mathbf{u}_h(t_k; \boldsymbol{\mu})\}_{k=0}^{N_t-1}$. Thus, providing the initial value to the LDM, along with the time-grid and the parametric input, the latent dynamics is solved over N_t steps, which, given the FOM discretizations in both test cases, accounts for a roll-out of size $N_t = 1000$. Similarly to the previous case, long-term predictions tests are carried out in a multi-query context by considering training, interpolation and extrapolation parameter instances. The results are reported in Table 2, again expressed in terms of the relative error indicator $\epsilon_{rel}(\mathbf{u}_h, \hat{\mathbf{u}}_h)$. In particular, considering the error averaged over the whole temporal interval $[0, T]$, it can be observed that the LDM's approximation error is below 10%, even in the worst case. Finally, in Table 2 (right), we report the CPU speed-ups achieved by the LDM framework compared to the FOM, for both test cases.
In Figure 6, we report the comparison between the FOM and LDM Burgers' equation solutions, for three different instances of the diffusivity parameter $\nu = 0.005$, $\nu = 0.1075$ and $\nu = 0.71$, respectively belonging to the training, interpolation and extrapolation sets. Being a long-term roll-out, the three solutions are computed over the complete temporal interval $[0, 30]$ with $N_t = 1000$. Specifically, the white horizontal lines highlight the end of the training temporal interval (solid) and of the validation interval (dotted), thus performing time-extrapolation for $t > T_2 = 15$. The third column of plots displays the relative error $\boldsymbol{\epsilon}_{rel}(t; \mathbf{u}_h, \hat{\mathbf{u}}_h, \boldsymbol{\mu})$ defined in Eq. (5.6). As observed, the highest spikes of the error occur towards the end of the time-extrapolation region, on the order of 10^{-2}. As illustrated on the right-most plot, showing the evolution of the latent state, the proposed fully-convolutional architecture effectively maintains spatial coherence between the high- and low-dimensional states. Indeed, the learned latent state evolution mirrors the high-dimensional evolution on a coarser grid, with a reduced dimension of $n = 16$, thus enhancing the model's interpretability, via a latent representation which preserves the key spatial features.

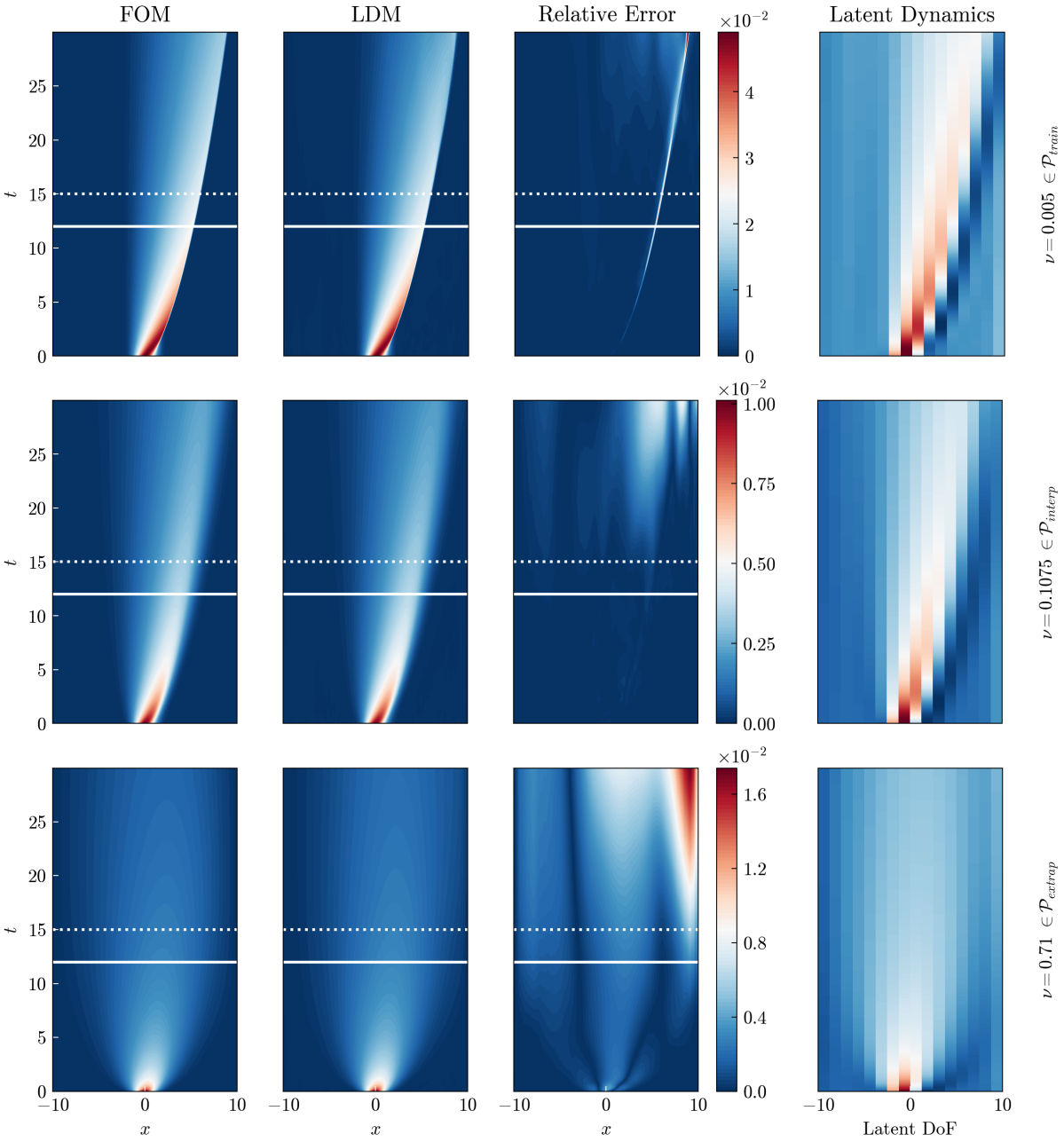

Figure 6: *Burgers' equation.* Evolution of the FOM solution (left), LDM solution (center left), relative error $\boldsymbol{\epsilon}_{rel}$ (center right), and of the latent state (right). The solutions are computed for three different values of the diffusivity parameter ν, respectively belonging to the training, interpolation and extrapolation sets. The horizontal white lines denote the end of the temporal training (solid) and validation (dotted) intervals, after which time-extrapolation is performed.

A comparison between ADR equation FOM and LDM solutions, for three testing parameter instances $\boldsymbol{\mu} = (0.02, 0.4, 0.511) \in \mathcal{P}_{train}$, $\boldsymbol{\mu} = (0.042, 0.433, 0.589) \in \mathcal{P}_{interp}$, $\boldsymbol{\mu} = (0.018, 0.414, 0.526) \in \mathcal{P}_{extrap}$, at the respective time instants $t = 30.433$, $t = 31.385$ and $t = 29.376$, all belonging to the time extrapolation interval ($t > T_2$), are shown in Figure 7. All three solutions refer to a $N_t = 1000$ steps roll-out on the complete temporal interval $[0, 10\pi]$. Again, the relative error indicator $\boldsymbol{\epsilon}_{rel}(t; \mathbf{u}_h, \hat{\mathbf{u}}_h, \boldsymbol{\mu})$ is reported, showing values on the order of 10^{-2}, mainly located in the region where the solution attains its maximum. Despite being more subtle in this 2D case, the latent state, showed in the right-most plots, still retains visual features that directly correspond to the high-dimensional solution. Specifically, the latent state

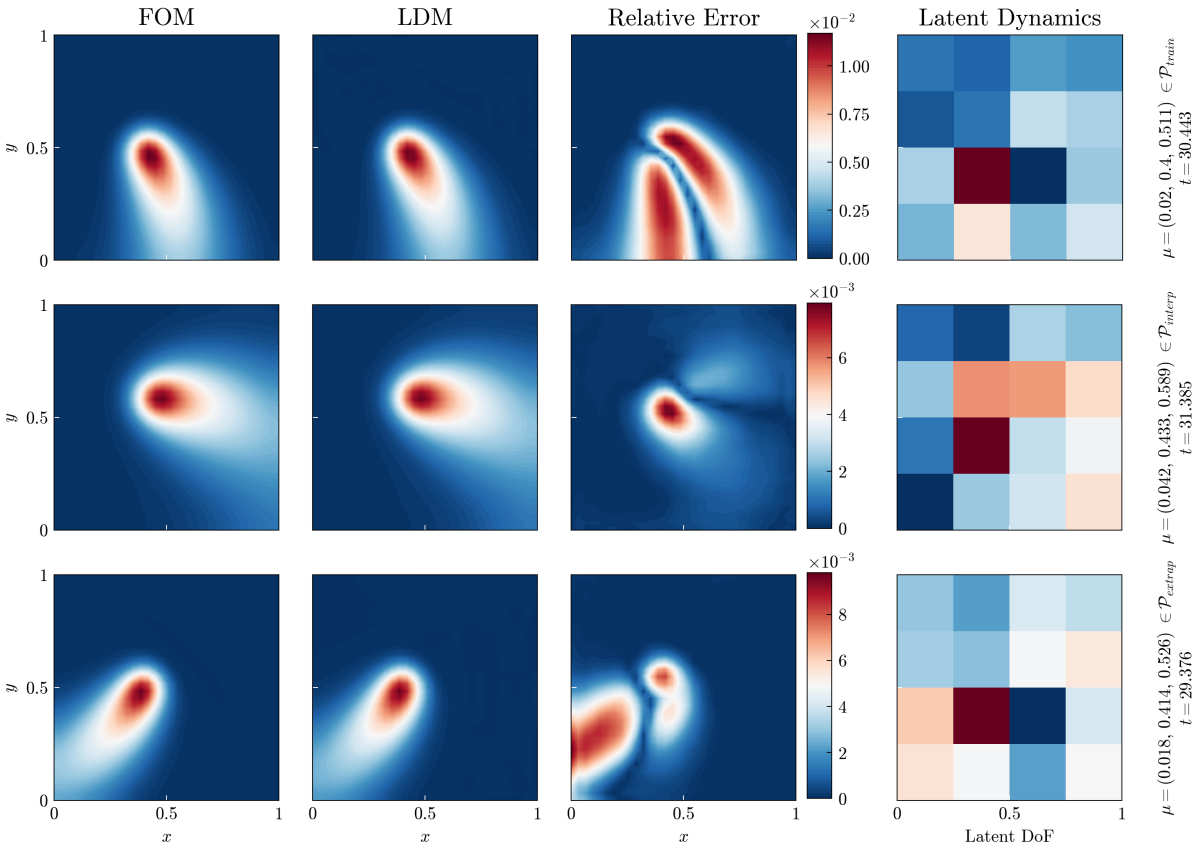

Figure 7: *ADR equation.* FOM solution (left), LDM solution (center left), relative error $\boldsymbol{\epsilon}_{rel}$ (center right), and latent state (right) for a fixed time instance t. The solutions are computed for three different instances of the parameter vector $\boldsymbol{\mu}$, respectively belonging to the training, interpolation and extrapolation sets. All the solutions refer to the time-extrapolation interval, being $t > T_2$.

components with large magnitude (indicated by red color) align with the position of the tail of the high-dimensional solution, on a coarser grid with $n = 16$.

5.2.2 Time-continuity

In this section, the numerical validation of the ROM time-continuous approximation property (Def. (2.1)) is addressed. The tests are performed for both Burgers' equation and ADR equation test cases, demonstrating the effectiveness of the LDM framework in providing a time-continuous approximation. Specifically, for each test case, we evaluate two models, with identical architectural details, but trained using different numerical methods for integrating the parameterized latent NODE, namely RK2 and RK4. In assessing time-continuity of the approximation, we focus on two main aspects: (i) the ability of the LDM to demonstrate zero-shot super-resolution with respect to temporal resolution, meaning that it can provide the same level of accuracy over different (finer) temporal discretizations that were not encountered during training (being the LDM trained on a fixed temporal discretization with Δt^*); and (ii) the validity of the upper bound (3.16), arising from the error decomposition which accounts for the numerical error sources, within a learnable context. To this mean, the relative error indicator $\epsilon_{rel}^{sup}(\mathbf{u}_h, \hat{\mathbf{u}}_h, \boldsymbol{\mu})$, defined in Eq. (5.7), is used to measure the maximum error over the time instants of the employed testing temporal discretization. The tests involve finer discretizations of the time interval $[0, T]$ on which the FOM is computed, with a number of steps up to $N_t = 10^4$ and $N_t = 3 \cdot 10^4$, for the Burgers' equation and ADR equation, respectively. The employed LDMs have been trained with time steps $\Delta t^* = 2\Delta t_{FOM}$ and $\Delta t^* = 5\Delta t_{FOM}$. Consequently, the tests are conducted on temporal discretizations that are up to $20\times$ and $50\times$ finer than those used during training. Figure 8 illustrates the behavior of the error indicator ϵ_{rel}^{sup} with respect to the testing temporal discretization step Δt.

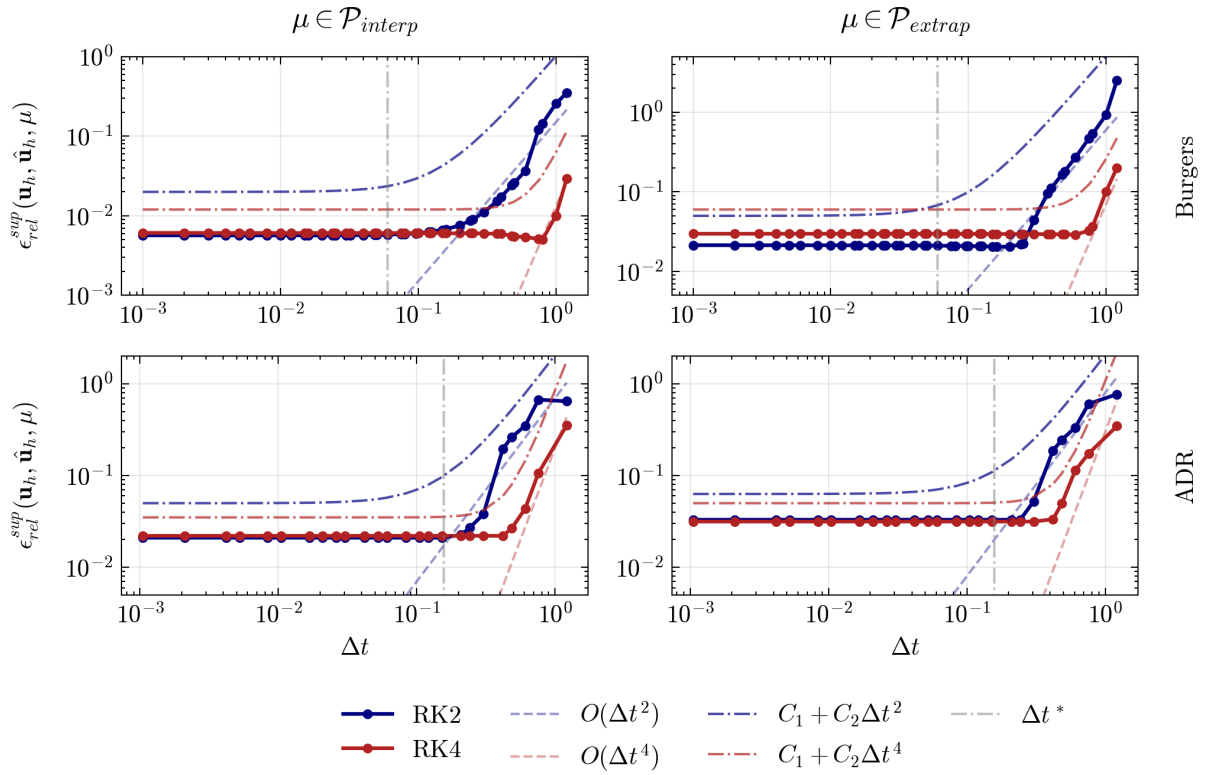

Figure 8: *Time-continuous approximation.* Assessment of the time-continuous approximation property for the two test cases: Burgers' equation (top), ADR equation (bottom). In both test cases, the property is evaluated using two integration schemes (RK2, RK4), and two different testing parameter instances (interpolation, extrapolation). Specifically, for the former test case, the adopted diffusivity coefficients are $\nu = 0.2075$ and $\nu = 0.61$. In the latter, $\boldsymbol{\mu} = (0.0283, 0.5, 0.4556)$ and $\boldsymbol{\mu} = (0.0337, 0.3754, 0.4767)$ are considered.

Specifically, the tests for both benchmarks problems are reported, with the two columns referring to two different interpolation and extrapolation testing parameter instances. In particular, the first row refers to the Burgers' equation test case, for the testing parameters instances $\nu = 0.2075 \in \mathcal{P}_{interp}$ and $\nu = 0.61 \in \mathcal{P}_{extrap}$. The second row refers to the ADR equation test case, for the testing parameter instances $\boldsymbol{\mu} = (0.0283, 0.5, 0.4556) \in \mathcal{P}_{interp}$ and $\boldsymbol{\mu} = (0.0337, 0.3754, 0.4767) \in \mathcal{P}_{extrap}$. The training temporal discretization step Δt^* is indicated, and a constant behavior of the error indicator is observed in all the test cases for $\Delta t \leq \Delta t^*$. It is evident that the resulting approximation provided by the LDM, in a discrete learnable context (ΔLDM_θ), can be queried at each time instance while preserving the training accuracy, with the time-continuous property holding also for the extrapolation parameter instances. Additionally, the validity of the upper bound, derived from the error decomposition formula (3.16), can be observed, with the error behaving as $O(\Delta t^p)$ for both $p = 2$ and $p = 4$, as the discretization becomes coarser.
Interestingly, the employment of a higher-order scheme, such as RK4, does not necessarily lead to improvements in terms of accuracy, primarily due to the models' capacity limitations. Rather, the region for which the model shows time-continuous property extends towards the right compared to lower-order methods like RK2, likely related to the fact that RK4 has a larger $\mathcal{A}$-stability region.

5.2.3 Zero-stability

Here, we aim to assess whether the zero-stability property, derived for the ΔLDM scheme in Section 3.2, holds in a learnable context. We test the framework's robustness with respect to initial value perturbations, by considering long-term predictions over the complete temporal interval $[0, T]$. Specifically, the experiment relies on the testing procedure outlined in Algorithm 4, with a randomly sampled Gaussian perturbation $\boldsymbol{\delta}_h \sim \mathcal{N}(0, \sigma^2 I_{N_h})$, where the amount of noise on $\mathbf{u}_{0,h}$ is progressively increased.

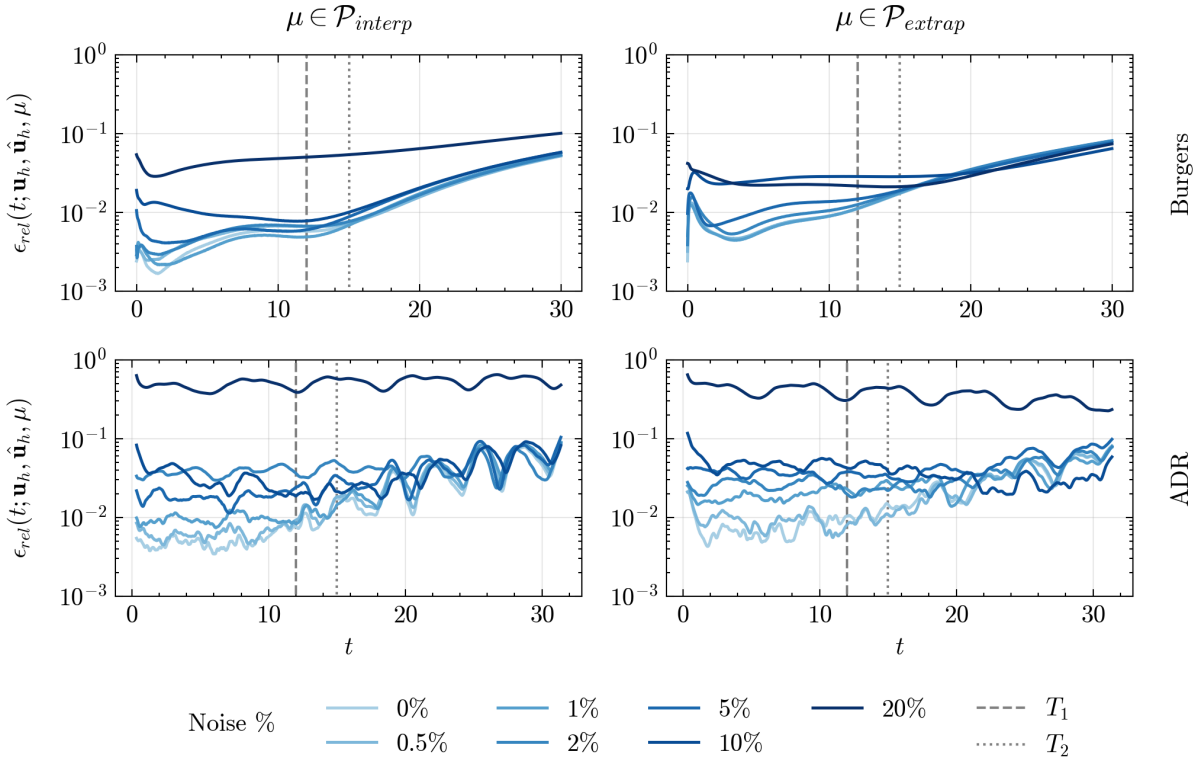

Figure 9: *Zero-stability.* Test of the zero-stability property for multiple initial value perturbation of increasing magnitude in the two test cases: Burgers' equation (top), ADR equation (bottom). The tests are conducted for two testing parameter instances from the interpolation and extrapolation sets, respectively. In the first test case, $\nu = 0.4775$ and $\nu = 0.6$ are considered, while in the second $\boldsymbol{\mu} = (0.035, 0.43, 0.41)$ and $\boldsymbol{\mu} = (0.05278, 0.4557, 0.3927)$ are used.

The time-dependent relative error indicator $\epsilon_{rel}(t; \mathbf{u}_h, \hat{\mathbf{u}}_h, \boldsymbol{\mu})$ (Eq. (5.5)) is employed for assessing the impact of the initial value perturbation, by benchmarking the perturbed LDM prediction directly against the FOM solution. In Figure 9, the relative error evolution is reported, for varying levels of noise, going from 0% up to 20%. The first row refers to the Burgers' equation test case, for the testing parameters instances $\nu = 0.4775 \in \mathcal{P}_{interp}$ and $\nu = 0.6 \in \mathcal{P}_{extrap}$, over the time interval $[0, 30]$. The second row illustrates the ADR equation test case, for the testing parameter instances $\boldsymbol{\mu} = (0.035, 0.43, 0.41) \in \mathcal{P}_{interp}$ and $\boldsymbol{\mu} = (0.05278, 0.4557, 0.3927) \in \mathcal{P}_{extrap}$, over the time interval $[0, 10\pi]$. The error between the FOM and the perturbed LDM approximation demonstrates a bounded behavior, confirming the zero-stability property, even when performing parameter extrapolation. Notably, when approaching the extrapolation interval ($t > T_2$), the errors for different levels of noise show a converging behavior, with an overall error magnitude of approximately 10^{-2}. In both test cases, the error remains bounded even as the perturbation magnitude reaches the 20% level, although it becomes significant.

5.2.4 Parameter dependence

Having introduced the LDM framework in a multi-query context, with a parameter-dependent formulation, we analyze the behavior of the LDM solution at extrapolation tasks with respect to the parameter input. We address two different aspects within the two benchmark problems. First, for Burgers' equation, we assess the impact of the sinusoidal encoding and the affine-modulation mechanism in the construction of the parameterized latent dynamics, proposed in Section 4.2.
The influence of two main factors is studied: the encoding dimension k and the capacity of the modulated convolutional NN employed within the latent dynamics. Our approach involves incrementally increasing the encoding size $k \in \{4, 8, 16\}$, alongside the number of affinely-modulated channels $n_c \in \{8, 16, 32\}$. The results, presented in Figure 10, illustrate how the parameter-dependent relative error $\epsilon_{rel}(\boldsymbol{\mu}; \mathbf{u}_h, \hat{\mathbf{u}}_h)$,

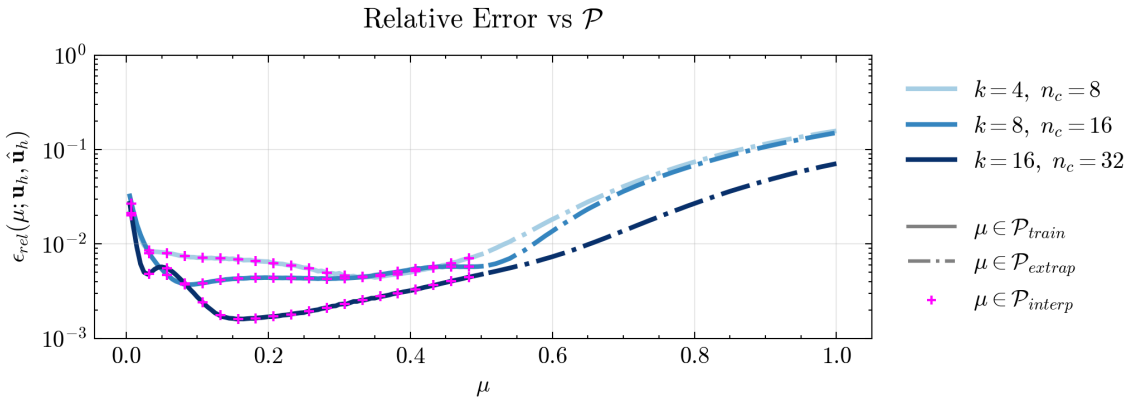

Figure 10: *Burgers' equation.* Behavior of the relative error indicator $\epsilon_{rel}(\boldsymbol{\mu}; \mathbf{u}_h, \hat{\mathbf{u}}_h)$ over the parameter space $\mathcal{P}$. The three curves refer to different model configurations with increasing encoding dimensions (k) and growing number of channels (n_c) in the latent dynamics' $\mathbf{f}_{n,\theta}$ modulated hidden layers. Continuous lines depict the error over the training instances ($\mathcal{P}_{train}$), while dash-dotted lines represent the error for the extrapolation instances ($\mathcal{P}_{extrap}$). Magenta crosses indicate a subset (for visualization purposes) of the interpolation instances ($\mathcal{P}_{interp}$).

defined in Eq. (5.4), varies with respect to the viscosity coefficient. As shown, the LDM employing a larger encoding dimension ($k = 16$) and a higher number of modulated channels ($n_c = 32$) achieves an error magnitude on the order 10^{-2} across the entire extrapolation interval $\nu \in (0.5, 1]$, exhibiting a more gradual increase compared to other configurations.

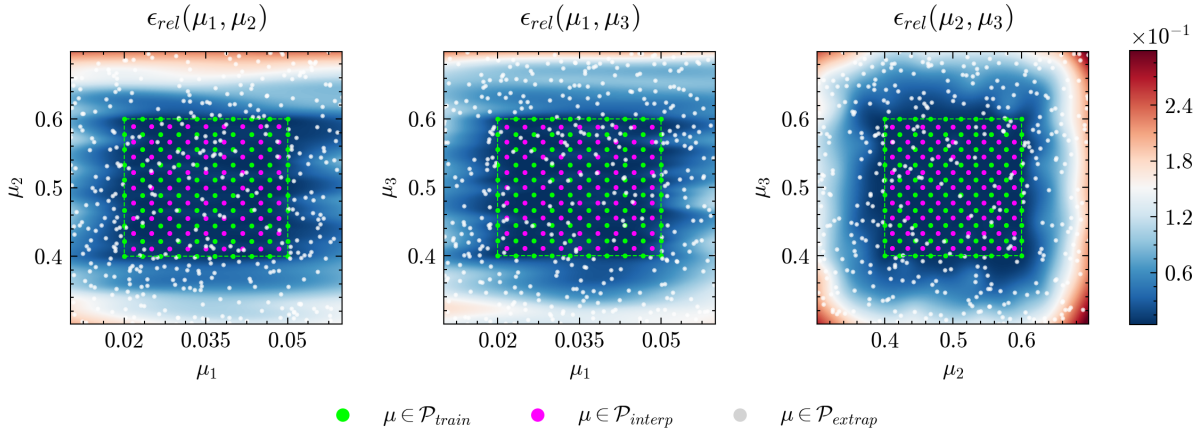

Figure 11: *ADR equation.* Behavior of the relative error indicator $\epsilon_{rel}(\boldsymbol{\mu}; \mathbf{u}_h, \hat{\mathbf{u}}_h)$ with respect to the three parameters component $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3)$ across the parameter space $\mathcal{P}$. Green points represent the training parameter instances ($\mathcal{P}_{train}$), while magenta points indicate interpolation testing instances ($\mathcal{P}_{interp}$). The white points correspond to extrapolation test instances ($\mathcal{P}_{extrap}$). Note that for pairs of parameters components (μ_i, μ_j) falling within the training interval, the remaining component μ_k is within the extrapolation range, allowing for extrapolation testing across all components $(i, j, k \in \{1, 2, 3\},\ i \neq j \neq k)$.

In the ADR test case, we focus on assessing the generalization capabilities for $n_\mu \geq 2$. The error behavior with respect to the different parameter components $\boldsymbol{\mu} = (\mu_1, \mu_2, \mu_3)$ is illustrated in Figure 11. The plots highlight the training region (green) including the training instances ($\mathcal{P}_{train}$), the interpolation instances ($\mathcal{P}_{interp}$). Being $n_\mu = 3$, the extrapolation capabilities with respect to each one of the components are tested. It is evident that the parameters related to the position of the source term (μ_2 and μ_3) have a major impact on the LDM prediction accuracy. As shown in the first and second plots, the error shows less variability with respect to the diffusivity parameter μ_1, whereas the third plot shows substantial increase in the error magnitude (order of 10^{-1}) as the boundaries of the parameter space are approached.

6 Conclusions

The concept of latent dynamics underlies many of the recently proposed DL-based ROMs addressing data-driven nonlinear order-reduction of time-dependent problems. This paradigm has been mainly driven by the adoption of recurrent architectures modeling the time-evolution of the reduced state within the latent space. However, RNN-based models, and more generally autoregressive ones, often require a temporal context, encoding current and previous high-fidelity FOM's snapshots, not allowing for the resulting ROM to be truly independent of the FOM in terms of online computational costs. Furthermore, these approaches are inherently tied to the temporal discretization employed during the offline training phase, limiting their ability to handle different testing temporal discretizations. Consequently, the learned dynamics cannot be interpreted in a continuous-sense, being constrained to a discrete temporal representation. As a result, further training or fine-tuning stages are required when encountering novel temporal discretizations, thereby entailing higher offline computational costs.
Neural ODEs (NODEs), coupled with autoencoders (AEs), present a promising alternative for constructing data-driven ROMs able to overcome these limitations. Recent studies have demonstrated the time-continuous approximation properties of NODEs in modeling dynamical systems, particularly when higher-order RK schemes are employed, guaranteeing the same error magnitude across refinements of temporal discretization. However, the time-continuous approximation capabilities of the AE-NODE architecture in multi-query settings had not been addressed prior to our research.
To this end, we extended the notion of *time-continuous approximation* towards a data-driven reduced order modeling context, and introduced the mathematical framework of *latent dynamics models* (LDMs) to formalize the AE-NODE architecture, and address its properties. Specifically, our framework develops along three level of approximation: time-continuous setting (LDM), time-discrete setting (ΔLDM), and learnable setting (ΔLDM_θ). Such chain of approximations enabled us to *(i)* analyze the error sources and derive stability estimates for the LDM approximation in a time continuous setting; *(ii)* address the numerical aspects of the method when integration schemes are employed for the solution of the latent dynamics in a time-discrete setting; *(iii)* derive a time-continuous approximation result in a learnable context (Theorem 3.5) when DNNs are employed to approximate the ΔLDM components. The resulting framework is characterized by an IVP-structure, requiring only the initial value to evolve the system, rather than a temporal context, leading to a ROM whose computational complexity is fully independent of the FOM's complexity. Additionally, the time-continuous approximation capabilities yield a ROM capable of providing the same level of accuracy for refined temporal discretizations relative to the training discretization.
From an architectural perspective, we propose adopting a fully convolutional structure for both the AE and the parameterized NODE. This approach enables the retention of spatial coherence at the latent level while substantially reducing the network size compared to classical AE-NODE dense architectures. Addressing the specific context of multi-query applications, we introduced a parameterized affine modulation mechanism to inject temporal and parametric information into the convolutional latent dynamics, enabling us to mitigate the additional costs typically associated with hypernetwork-based parameterization approaches. The numerical and time-continuous approximation properties of the proposed architecture have been tested on high-dimensional dynamical systems arising from the semi-discretization of parameterized nonlinear time-dependent PDEs.
While this work has primarily focused on the temporal aspects of the proposed framework, its scalability with respect to the spatial discretization needs further investigation. Examining the generalization capabilities and computational efficiency across varying levels of spatial resolution is essential for extending the framework's applicability to large-scale problems. In particular, possible future directions may consider coupling the LDM framework with GNNs to enhance its capability of handling complex domains and three-dimensional geometries. Additionally, the recent advancements in the field of operator learning can be leveraged to define an LDM architecture with continuous approximation capabilities in both the temporal and the spatial domains.

References

[1] R.A. Adams and J.J.F. Fournier. *Sobolev Spaces*. Academic Press, 2003.

[2] Francesco Andreuzzi, Nicola Demo, and Gianluigi Rozza. A dynamic mode decomposition extension for the forecasting of parametric dynamical systems. *SIAM Journal on Applied Dynamical Systems*, 22(3):2432–2458, 2023.

[3] V.I. Arnold and R.A. Silverman. *Ordinary Differential Equations*. MIT Press, 1978.

[4] Ibrahim Ayed, Emmanuel de Bézenac, Arthur Pajot, Julien Brajard, and Patrick Gallinari. Learning dynamical systems from partial observations, 2019.

[5] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.

[6] Peter Benner, Serkan Gugercin, and Karen Willcox. A survey of projection-based model reduction methods for parametric dynamical systems. *SIAM Review*, 57(4):483–531, 2015.

[7] Kaushik Bhattacharya, Bamdad Hosseini, Nikola B Kovachki, and Andrew M Stuart. Model reduction and neural networks for parametric PDEs. *SMAI J. Comput. Mat.*, 7:121–157, 2021.

[8] Christophe Bonneville, Xiaolong He, April Tran, Jun Sur Park, William Fries, Daniel A Messenger, Siu Wun Cheung, Yeonjong Shin, David M Bortz, Debojyoti Ghosh, et al. A comprehensive review of latent space dynamics identification algorithms for intrusive and non-intrusive reduced-order-modeling. *arXiv preprint arXiv:2403.10748*, 2024.

[9] Johannes Brandstetter, Daniel E. Worrall, and Max Welling. Message passing neural PDE solvers. In *International Conference on Learning Representations*, 2022.

[10] Simone Brivio, Stefania Fresca, Nicola Rares Franco, and Andrea Manzoni. Error estimates for POD-DL-ROMs: a deep learning framework for reduced order modeling of nonlinear parametrized PDEs enhanced by proper orthogonal decomposition. *Adv. Comput. Math.*, 50(33), 2024.

[11] Simone Brivio, Stefania Fresca, and Andrea Manzoni. PTPI-DL-ROMs: pre-trained physics-informed deep learning-based reduced order models for nonlinear parametrized PDEs. *arXiv preprint arXiv:2405.08558*, 2024.

[12] Steven L Brunton, Bingni W Brunton, Joshua L Proctor, and J Nathan Kutz. Koopman invariant subspaces and finite linear representations of nonlinear dynamical systems for control. *PloS one*, 11(2):e0150171, 2016.

[13] Steven L. Brunton, Joshua L. Proctor, and J. Nathan 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.

[14] Kathleen Champion, Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Data-driven discovery of coordinates and governing equations. *Proceedings of the National Academy of Sciences*, 116(45):22445–22451, 2019.

[15] Saifon Chaturantabut and Danny C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.

[16] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[17] Xing Chen, Flavio Abreu Araujo, Mathieu Riou, Jacob Torrejon, Dafiné Ravelosona, Wang Kang, Weisheng Zhao, Julie Grollier, and Damien Querlioz. Forecasting the outcome of spintronic experiments with neural ordinary differential equations. *Nature Communications*, 13(1), February 2022.

[18] L. Cicci, S. Fresca, S. Pagani, A. Manzoni, and A. Quarteroni. Projection-based reduced order models for parameterized nonlinear time-dependent problems arising in cardiac mechanics. *Mathematics in Engineering*, 5(2):1–38, 2023.

[19] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus), 2016.

[20] Paolo Conti, Giorgio Gobat, Stefania Fresca, Andrea Manzoni, and Attilio Frangi. Reduced order modeling of parametrized systems through autoencoders and sindy approach: continuation of periodic solutions. *Computer Methods in Applied Mechanics and Engineering*, 411:116072, 2023.

[21] Paolo Conti, Mengwu Guo, Andrea Manzoni, and Jan S. Hesthaven. Multi-fidelity surrogate modeling using long short-term memory networks. *Computer Methods in Applied Mechanics and Engineering*, 404:115811, February 2023.

[22] R. Courant, K. Friedrichs, and H. Lewy. Über die partiellen Differenzengleichungen der mathematischen Physik. *Mathematische Annalen*, 100:32–74, January 1928.

[23] Niccolo Dal Santo, Simone Deparis, Andrea Manzoni, and Alfio Quarteroni. An algebraic least squares reduced basis method for the solution of nonaffinely parametrized stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 344:186–208, 2019.

[24] Prafulla Dhariwal and Alex Nichol. Diffusion models beat GANs on image synthesis, 2021.

[25] Domenico Di Sante, Matija Medvidović, Alessandro Toschi, Giorgio Sangiovanni, Cesare Franchini, Anirvan M. Sengupta, and Andrew J. Millis. Deep learning the functional renormalization group. *Physical Review Letters*, 129(13), September 2022.

[26] Franck Djeumou, Cyrus Neary, Eric Goubault, Sylvie Putot, and Ufuk Topcu. Neural networks with physics-informed architectures and constraints for dynamical systems modeling. In *Learning for Dynamics and Control Conference*, pages 263–277. PMLR, 2022.

[27] Junming Duan and Jan S. Hesthaven. Non-intrusive data-driven reduced-order modeling for time-dependent parametrized problems. *Journal of Computational Physics*, 497:112621, 2024.

[28] Weinan E. A proposal on machine learning via dynamical systems. *Communications in Mathematics and Statistics*, 5(1):1–11, 3 2017. Dedicated to Professor Chi-Wang Shu on the occasion of his 60th birthday.

[29] Nicola Farenga, Stefania Fresca, and Andrea Manzoni. Neural latent dynamics models. In *The Symbiosis of Deep Learning and Differential Equations II, NeurIPS*, 2022.

[30] Nicola Franco, Andrea Manzoni, and Paolo Zunino. A deep learning approach to reduced order modelling of parameter dependent partial differential equations. *Mathematics of Computation*, 92(340):483–524, 2023.

[31] Nicola Rares Franco, Stefania Fresca, Filippo Tombari, and Andrea Manzoni. Deep learning-based surrogate models for parametrized PDEs: Handling geometric variability through graph neural networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 33(12):123121, 12 2023.

[32] Nicola Rares Franco, Andrea Manzoni, and Paolo Zunino. Mesh-informed neural networks for operator learning in finite element spaces, 2023.

[33] Stefania Fresca, Luca Dede', and Andrea Manzoni. A comprehensive deep learning-based approach to reduced order modeling of nonlinear time-dependent parametrized PDEs. *Journal of Scientific Computing*, 87:1–36, 2021.

[34] Stefania Fresca, Federico Fatone, and Andrea Manzoni. Long-time prediction of nonlinear parametrized dynamical systems by deep learning-based reduced order models. *Mathematics in Engineering*, 5(6):1–36, 2023.

[35] Stefania Fresca and Andrea Manzoni. POD-DL-ROM: Enhancing deep learning-based reduced order models for nonlinear parametrized PDEs by proper orthogonal decomposition. *Computer Methods in Applied Mechanics and Engineering*, 388:114181, 2022.

[36] William D Fries, Xiaolong He, and Youngsoo Choi. LaSDI: Parametric latent space dynamics identification. *Computer Methods in Applied Mechanics and Engineering*, 399:115436, 2022.

[37] Kai Fukami, Takaaki Murata, Kai Zhang, and Koji Fukagata. Sparse identification of nonlinear dynamics with low-dimensionalized flow representations. *Journal of Fluid Mechanics*, 926:A10, 2021.

[38] Lawson Fulton, Vismay Modi, David Duvenaud, David I. W. Levin, and Alec Jacobson. Latent-space dynamics for reduced deformable simulation. *Computer Graphics Forum*, 38(2):379–391, 2019.

[39] Nicholas Geneva and Nicholas Zabaras. Modeling the dynamics of PDE systems with physics-constrained deep auto-regressive networks. *Journal of Computational Physics*, 403:109056, February 2020.

[40] Felix A. Gers, Jürgen Schmidhuber, and Fred Cummins. Learning to forget: Continual prediction with lstm. *Neural Computation*, 12(10):2451–2471, 2000.

[41] Arnab Ghosh, Harkirat Behl, Emilien Dupont, Philip Torr, and Vinay Namboodiri. Steer : Simple temporal regularization for neural ode. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 14831–14843. Curran Associates, Inc., 2020.

[42] Laurent Girin, Simon Leglaive, Xiaoyu Bie, Julien Diard, Thomas Hueber, and Xavier Alameda-Pineda. Dynamical variational autoencoders: A comprehensive review. *arXiv preprint arXiv:2008.12595*, 2020.

[43] Francisco J Gonzalez and Maciej Balajewicz. Deep convolutional recurrent autoencoders for learning low-dimensional feature dynamics of fluid systems. *arXiv preprint arXiv:1808.01346*, 2018.

[44] Constantin Greif and Karsten Urban. Decay of the kolmogorov n-width for wave problems, 2019.

[45] Jayesh K. Gupta and Johannes Brandstetter. Towards multi-spatiotemporal-scale generalized PDE modeling, 2022.

[46] Ingo Gühring and Mones Raslan. Approximation rates for neural networks with encodable weights in smoothness spaces. *Neural Networks*, 134:107–130, 2021.

[47] David Ha, Andrew Dai, and Quoc V. Le. Hypernetworks, 2016.

[48] Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International conference on machine learning*, pages 2555–2565. PMLR, 2019.

[49] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 630–645. Springer, 2016.

[50] P. Henrici. *Discrete Variable Methods in Ordinary Differential Equations*. Wiley, 1962.

[51] Jan S. Hesthaven, Cecilia Pagliantini, and Gianluigi Rozza. Reduced basis methods for time-dependent problems. *Acta Numerica*, 31:265–345, 2022.

[52] Jan S. Hesthaven and Stefano Ubbiali. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *J. Comput. Phys.*, 363:55–78, 2018.

[53] Geoffrey E Hinton and Richard Zemel. Autoencoders, minimum description length and helmholtz free energy. *Advances in neural information processing systems*, 6, 1993.

[54] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[55] Zijie Huang, Yizhou Sun, and Wei Wang. Learning continuous system dynamics from irregularly-sampled partial observations. *Advances in Neural Information Processing Systems*, 33:16177–16187, 2020.

[56] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

[57] J. Nagoor Kani and Ahmed H. Elsheikh. DR-RNN: A deep residual recurrent neural network for model reduction, 2017.

[58] Patrick Kidger. On neural differential equations, 2022.

[59] Youngkyu Kim, Youngsoo Choi, David Widemann, and Tarek Zohdi. A fast and accurate physics-informed neural network reduced order model with shallow masked autoencoder. *Journal of Computational Physics*, 451:110841, February 2022.

[60] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

[61] Ivica Kičić, Pantelis R. Vlachas, Georgios Arampatzis, Michail Chatzimanolakis, Leonidas Guibas, and Petros Koumoutsakos. Adaptive learning of effective dynamics for online modeling of complex systems. *Computer Methods in Applied Mechanics and Engineering*, 415:116204, 2023.

[62] A. Kolmogoroff. Uber die beste annaherung von funktionen einer gegebenen funktionenklasse. *Annals of Mathematics*, 37(1):107–110, 1936.

[63] Katiana Kontolati, Somdatta Goswami, George Em Karniadakis, and Michael D. Shields. Learning nonlinear operators in latent spaces for real-time predictions of complex dynamics in physical systems. *Nature Communications*, 15(1):5101, Jun 2024.

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

[65] Nikola Kovachki, Zongyi Li, Burigede Liu, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Neural operator: Learning maps between function spaces with applications to PDEs. *Journal of Machine Learning Research*, 24(89):1–97, 2023.

[66] Aditi S. Krishnapriyan, Alejandro F. Queiruga, N. Benjamin Erichson, and Michael W. Mahoney. Learning continuous models for continuous physics, 2023.

[67] Andrea Lario, Romit Maulik, Oliver T. Schmidt, Gianluigi Rozza, and Gianmarco Mengaldo. Neural-network learning of spod latent dynamics. *Journal of Computational Physics*, 468:111475, 2022.

[68] Michele Lazzara, Max Chevalier, Corentin Lapeyre, and Olivier Teste. Neuralode-based latent trajectories into autoencoder architecture for surrogate modelling of parametrized high-dimensional dynamical systems. In *Artificial Neural Networks and Machine Learning – ICANN 2023*, pages 497–508, Cham, 2023. Springer Nature Switzerland.

[69] Kookjin Lee and Kevin T Carlberg. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *Journal of Computational Physics*, 404:108973, 2020.

[70] Kookjin Lee and Eric J. Parish. Parameterized neural ordinary differential equations: applications to computational physics problems. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 477(2253), September 2021.

[71] Christian Møldrup Legaard, Thomas Schranz, Gerald Schweiger, Ján Drgoňa, Basak Falay, Cláudio Gomes, Alexandros Iosifidis, Mahdi Abkar, and Peter Gorm Larsen. Constructing neural network-based models for simulating dynamical systems, 2022.

[72] Zijie Li, Saurabh Patil, Francis Ogoke, Dule Shu, Wilson Zhen, Michael Schneier, John R. Buchanan Jr., and Amir Barati Farimani. Latent neural PDE solver: a reduced-order modelling framework for partial differential equations, 2024.

[73] Alec J. Linot, Joshua W. Burby, Qi Tang, Prasanna Balaprakash, Michael D. Graham, and Romit Maulik. Stabilized neural ordinary differential equations for long-time forecasting of dynamical systems. *Journal of Computational Physics*, 474:111838, 2023.

[74] Phillip Lippe, Bas Veeling, Paris Perdikaris, Richard Turner, and Johannes Brandstetter. PDE-refiner: Achieving accurate long rollouts with neural PDE solvers. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 67398–67433. Curran Associates, Inc., 2023.

[75] Yiping Lu, Aoxiao Zhong, Quanzheng Li, and Bin Dong. Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3276–3285. PMLR, 10–15 Jul 2018.

[76] Emmanuel Menier, Sebastian Kaltenbach, Mouadh Yagoubi, Marc Schoenauer, and Petros Koumoutsakos. Interpretable learning of effective dynamics for multiscale systems. *arXiv preprint arXiv:2309.05812*, 2023.

[77] Nikolaj T. Mücke, Sander M. Bohté, and Cornelis W. Oosterlee. Reduced order modeling for parameterized time-dependent PDEs using spatially and memory aware deep learning. *J. Comput. Sci.*, 53:101408, 2021.

[78] Federico Negri, Andrea Manzoni, and David Amsallem. Efficient model reduction of parametrized systems by matrix discrete empirical interpolation. *Journal of Computational Physics*, 303:431–454, 2015.

[79] Alex Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models, 2021.

[80] Monica Nonino, Francesco Ballarin, Gianluigi Rozza, and Yvon Maday. A reduced basis method by means of transport maps for a fluid–structure interaction problem with slowly decaying kolmogorov n-width. *Advances in Computational Science and Engineering*, 1(1):36–58, 2023.

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

[82] Pranshu Pant, Ruchit Doshi, Pranav Bahl, and Amir Barati Farimani. Deep learning for reduced order modelling and efficient temporal evolution of fluid simulations. *Phys. Fluids*, 33(10):107101, 2021.

[83] Benjamin Peherstorfer. Breaking the kolmogorov barrier with nonlinear model reduction. *Notices of the American Mathematical Society*, 69(5):725–733, 2022.

[84] Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. Film: Visual reasoning with a general conditioning layer, 2017.

[85] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W. Battaglia. Learning mesh-based simulation with graph networks, 2021.

[86] A. Quarteroni, A. Manzoni, and F. Negri. *Reduced Basis Methods for Partial Differential Equations: An Introduction*. UNITEXT. Springer International Publishing, 2015.

[87] Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. *Numerical Mathematics*. Springer New York, 2007.

[88] Alfio Quarteroni and Alberto Valli. *Numerical approximation of partial differential equations*. Springer Series in Computational Mathematics. Springer, Berlin, Germany, 1 edition, September 2008.

[89] Alejandro F. Queiruga, N. Benjamin Erichson, Dane Taylor, and Michael W. Mahoney. Continuous-in-depth neural networks, 2020.

[90] Anthony Ralston. Runge-kutta methods with minimum error bounds. *Mathematics of Computation*, 16(80):431–437, 1962.

[91] Francesco Romor, Giovanni Stabile, and Gianluigi Rozza. Non-linear manifold reduced-order models with convolutional autoencoders and reduced over-collocation method. *Journal of Scientific Computing*, 94(3):74, 2023.

[92] Yulia Rubanova, Ricky T. Q. Chen, and David K Duvenaud. Latent ordinary differential equations for irregularly-sampled time series. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[93] Omer San, Romit Maulik, and Mansoor Ahmed. An artificial neural network framework for reduced order modeling of transient flows. *Comm. Nonlin. Sci. Numer. Simul.*, 77:271–287, 2019.

[94] Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of fluid mechanics*, 656:5–28, 2010.

[95] Aleksei Sholokhov, Yuying Liu, Hassan Mansour, and Saleh Nabi. Physics-informed neural ode (pinode): embedding physics into models using collocation points. *Scientific Reports*, 13(1):10166, Jun 2023.

[96] Vidar Thomee. *Galerkin finite element methods for parabolic problems*. Springer Series in Computational Mathematics. Springer, Berlin, Germany, 2 edition, July 2006.

[97] Matteo Torzoni, Andrea Manzoni, and Stefano Mariani. A multi-fidelity surrogate model for structural health monitoring exploiting model order reduction and artificial neural networks. *Mechanical Systems and Signal Processing*, 197:110376, 2023.

[98] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[99] Pantelis R Vlachas, Georgios Arampatzis, Caroline Uhler, and Petros Koumoutsakos. Multiscale simulations of complex systems by learning their effective dynamics. *Nature Machine Intelligence*, 4(4):359–366, 2022.

[100] Pantelis R. Vlachas, Wonmin Byeon, Zhong Y. Wan, Themistoklis P. Sapsis, and Petros Koumoutsakos. Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2213):20170844, 2018.

[101] Pantelis R Vlachas, Julija Zavadlav, Matej Praprotnik, and Petros Koumoutsakos. Accelerated simulations of molecular systems through learning of effective dynamics. *Journal of Chemical Theory and Computation*, 18(1):538–549, 2021.

[102] Pantelis-Rafail Vlachas, Jaideep Pathak, Brian R Hunt, Themistoklis P Sapsis, Michelle Girvan, Edward Ott, and Petros Koumoutsakos. Backpropagation algorithms and reservoir computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics. *Neural Networks*, 126:191–217, 2020.

[103] Qian Wang, Jan S. Hesthaven, and Deep Ray. Non-intrusive reduced order modeling of unsteady flows using artificial neural networks with application to a combustion problem. *Journal of Computational Physics*, 384:289–307, 2019.

[104] Qian Wang, Nicolò Ripamonti, and Jan S. Hesthaven. Recurrent neural network closure of parametric POD-galerkin reduced-order models based on the mori-zwanzig formalism. *J. Comput. Phys.*, 410:109402, 2020.

[105] Z. Wang, D. Xiao, F. Fang, R. Govindan, C. C. Pain, and Y. Guo. Model identification of reduced order fluid dynamics systems using deep learning. *International Journal for Numerical Methods in Fluids*, 86(4):255–268, 2018.

[106] Manuel Watter, Jost Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. *Advances in neural information processing systems*, 28, 2015.

[107] Tianshu Wen, Kookjin Lee, and Youngsoo Choi. Reduced-order modeling for parameterized PDEs via implicit neural representations. *arXiv preprint arXiv:2311.16410*, 2023.

[108] S. Wiewel, M. Becher, and N. Thuerey. Latent space physics: Towards learning the temporal evolution of fluid flow. *Computer Graphics Forum*, 38(2):71–82, 2019.

[109] Tailin Wu, Takashi Maruyama, and Jure Leskovec. Learning to accelerate partial differential equations via latent global evolution. *Advances in Neural Information Processing Systems*, 35:2240–2253, 2022.

[110] Yuan Yin, Matthieu Kirchmeyer, Jean-Yves Franceschi, Alain Rakotomamonjy, and Patrick Gallinari. Continuous PDE dynamics forecasting with implicit neural representations. *arXiv preprint arXiv:2209.14855*, 2022.

A Continuous-depth and continuous-time paradigms

In the context of deep learning, the notion of *continuous* architectures has been recently introduced. In particular, we have to make a distinction between the, strictly related, concepts of *continuous-depth* and *continuous-time* models.

Continuous-depth. *Continuous-depth* architectures [89], likewise *neural ordinary differential equations* [16], have been introduced by means of continuous limits of *residual neural networks*, by considering the correspondence between the ResNets' residual formulation and forward Euler scheme [75]. In particular, by considering the general case where we have to model input-output relations, considering feature-target pairs (x, y), a ResNet with L layers takes the form

$$x \mapsto h_1 \mapsto h_2 \mapsto \cdots \mapsto h_L \mapsto \hat{y}, \tag{A.1}$$

where $\mapsto$ are residual blocks of the form $h_{l+1} = h_l + f_\theta(h_l, l)$, so the whole network reads as

$$\begin{cases} \hat{y} = h_{L+1}, \\ h_{l+1} = h_l + f_\theta(h_l, l), \quad l = 0, \dots, L, \\ h_0 = x, \end{cases}$$

where $f_\theta(\cdot, l)$ represents the inner learnable block of weights θ of the l-th layer. In this view, the discrete scheme outlined above may be seen as a discretization, by means of the explicit Euler scheme with $\Delta t = 1$, of a continuous problem on $[0, T]$, reading as

$$\begin{cases} \hat{y} = h(T), \\ \dot{h}(t) = f_\theta(t, h(t)), \quad t \in (0, T), \\ h(0) = x. \end{cases}$$

Thus, via the above continuous formulation, any numerical method for the solution of ODEs can now be employed for integrating the network $f_\theta(t, h(t))$, and retrieving the final output $y = h(T)$, where $h(t)$ is now a continuously evolving hidden-state. The arbitrariness of the employed discretization of the integration interval leads to the concept of *continuous-depth*, since, in this context, employing a finer time-discretization would mean taking a larger number of function evaluations within the numerical solver, i.e., a higher number of intermediate hidden states, naively corresponding to a *deeper* residual architecture. Thus, the continuous-depth nature of such architecture resides in: (i) the continuous nature of the variable now indexing the hidden layers $t \in \mathbb{R}$, rather than discrete indexes $l \in \mathbb{N}$, (ii) and in their continuous evolution, by construction. We remark that, in this case, we are not interested in the evolution of the hidden state $h(t)$, but only in the final output $\hat{y} = h(T)$. Moreover, the fact that the hidden state now evolves continuously accordingly to an ODE, means that the relation $x = h(0) \mapsto h(T) = y$ is being modeled by a homeomorphism, i.e., a one-to-one continuous mapping with continuous inverse.

Continuous-time. The dynamical nature of continuous-depth architectures enables their employment in the context of time-series modeling, leading to the concept of *continuous-time* models, to model relations between input-target pairs of the form $(x, y) = (x_0, \{x_1, ..., x_n\})$ in a *recurrent* manner, where the target is made by n sampled steps of a continuous trajectory over $[0, T]$. In this setting, the sampled state $x(t)$ can be modeled continuously by means of $\hat{x}(t)$, evolving accordingly to the following *neural* ODE

$$\begin{cases} \dot{\hat{x}}(t) = f_\theta(t, h(t)), \quad t \in (0, T), \\ \hat{x}(0) = x_0. \end{cases} \tag{A.2}$$

where f_θ is the neural network parameterizing its dynamics. In particular, (A.2) can be seen as a continuous formulation of a recurrent neural network, modeling the evolution of $x(t)$. In such setting, since the whole trajectory is now the target, the objective being minimized is $\sum_{i=1}^{n} \|x(t_i) - \hat{x}(t_i)\|^2$, meaning that the prediction is being unrolled over the discretized interval, as in recurrent settings.

B Notation

B.1 Norms

Throughout the paper we indicate by $\|\cdot\|$ the Euclidean norm, typically employed in either $\mathbb{R}^{N_h}$ or $\mathbb{R}^n$, for vectors lying in the high-dimensional space (e.g. $\mathbf{u}_h, \hat{\mathbf{u}}_h$) or in the latent space (e.g. $\mathbf{u}_n$), respectively. Regarding functional spaces, we rely on Sobolev spaces $W^{k,p}(\Omega)$, and specifically on their norms, to measure distances both *(i)* in the introduced minimization problems, and *(ii)* in the approximation result provided in Theorem 3.5. In the following their definition, accordingly to [1], is provided.

Definition B.1. Let $\Omega \subset \mathbb{R}^d$ be a domain, $1 \leq p \leq \infty$, $k \in \mathbb{N}$, the Sobolev space $W^{k,p}(\Omega;\mathbb{R})$ is defined as

$$W^{k,p}(\Omega;\mathbb{R}) = \left\{ f : \Omega \to \mathbb{R} \ : \ D^\alpha f \in L^p(\Omega), \ \forall \alpha \in \mathbb{N}_0^d, \ |\alpha| \leq k \right\},$$

equipped with the following norm

$$\|f\|_{W^{k,p}(\Omega;\mathbb{R})} := \begin{cases} \left(\sum_{|\alpha|\leq k} \|D^\alpha f\|^p_{L^p(\Omega)} \right)^{1/p}, & 1 \leq p < \infty, \\ \max_{|\alpha|\leq k} \|D^\alpha f\|_{L^\infty(\Omega)}, & p = \infty, \end{cases}$$

where D^α indicates the weak (distributional) derivative of order α.

B.2 Symbols

For the sake of readability, in the following, we report a table summarizing the notation employed within the manuscript.

Symbol	Description
$\mathbf{u}_h(t;\boldsymbol{\mu})$	Full-order model state
$\mathbf{f}_h(t,\mathbf{u}_h(t;\boldsymbol{\mu});\boldsymbol{\mu})$	Full-order model vector field
$\tilde{\mathbf{u}}_h(t;\boldsymbol{\mu})$	LDM full-order state approximation
$\mathbf{u}_n(t;\boldsymbol{\mu})$	LDM latent state
$\mathbf{f}_n(t,\mathbf{u}_n(t;\boldsymbol{\mu});\boldsymbol{\mu})$	LDM latent vector field
$\tilde{\mathbf{z}}_h(t;\boldsymbol{\mu})$	Perturbed LDM full-order state approximation
$\mathbf{z}_n(t;\boldsymbol{\mu})$	Perturbed LDM latent state
$\tilde{\mathbf{u}}_h^k(\boldsymbol{\mu})$	ΔLDM full-order state approximation at t_k
$\mathbf{u}_n^k(\boldsymbol{\mu})$	ΔLDM latent state at t_k
$\tilde{\mathbf{z}}_h^k(\boldsymbol{\mu})$	Perturbed ΔLDM full-order state approximation at t_k
$\mathbf{z}_n^k(\boldsymbol{\mu})$	Perturbed ΔLDM latent state at t_k
$\hat{\mathbf{u}}_h(t;\boldsymbol{\mu})$	Learned LDM$_\theta$ full-order state approximation
$\hat{\mathbf{u}}_n(t;\boldsymbol{\mu})$	Learned LDM$_\theta$ latent state
$\hat{\mathbf{u}}_h^k(\boldsymbol{\mu})$	Learned ΔLDM$_\theta$ full-order state approximation at t_k
$\hat{\mathbf{u}}_n^k(\boldsymbol{\mu})$	Learned ΔLDM$_\theta$ latent state at t_k
$\mathbf{f}_{n,\theta}(t,\ \cdot\ ;\boldsymbol{\mu})$	Learned (Δ)LDM$_\theta$ latent vector field

Abstract in lingua italiana

Tale lavoro introduce il framework matematico dei *latent dynamics models* (LDM) (modelli a dinamica latente) per la riduzione d'ordine di equazioni alle derivate parziali (PDEs) nonlineari, caratterizzate da dipendenza temporale e parametrica. Il framework reinterpreta la riduzione della dimensionalità nonlineare in un contesto dinamico, vincolando lo stato latente a evolvere secondo un'equazione differenziale ordinaria (ODE). Il framework si sviluppa su tre livelli di approssimazione. In primo luogo viene adottata una formulazione continua in tempo (LDM), per la quale viene riportato un risultato di stabilità. Successivamente, il caso discreto viene studiato, analizzando l'utilizzo di schemi Runge-Kutta espliciti nella risoluzione numerica del problema latente, portando alla formulazione ΔLDM. In un contesto di apprendimento automatico (ΔLDM_θ), in cui reti neurali approssimano le componenti del ΔLDM, viene dimostrato un risultato di approssimazione.
In ambito Deep Learning, l'architettura Neural ODE parametrizzata è estesa al caso convoluzionale, in cui l'informazione dei parametri di input è gestita tramite un meccanismo di modulazione affine. Un'architettura convoluzionale in grado di mantenere coerenza spaziale è adottata per la costruzione dell'autoencoder, migliorando così l'interpretabilità a livello latente.
Le capacità del framework di ottenere un'approssimazione continua in tempo, nel caso parametrizzato, sono testate su due problemi: *(i)* equazione di Burgers (nonlineare) monodimensionale, *(ii)* equazione di trasporto-reazione-diffusione bidimesionale, caratterizzata da una dipendenza parametrica nonaffine. I risultati evidenziano il notevole potenziale dei LDM nel garantire un livello di accuratezza predefinito per discretizzazioni temporali più fini rispetto a quella adottata nella fase di apprendimento offline, rappresentando quindi un framework rigoroso per la riduzione d'ordine in un contesto continuo in tempo.

Parole chiave: Modelli di ordine ridotto, dinamica latente, apprendimento automatico

Acknowledgements

This work is dedicated to the memory of Pietro De Bernardini, a brilliant mind with whom I had the honor of beginning this journey.

Thanks to my family for having taught me the value of work and dedication.

Thanks to my advisors for their patience and guidance.