Chapter 8

Neural Galerkin schemes for sequential-in-time solving of partial differential equations with deep networks

Jules Berman, Paul Schwerdtner, and Benjamin Peherstorfer*

Courant Institute of Mathematical Sciences, New York University, New York, NY, United States

**Corresponding author: e-mail address: peghersto@cims.nyu.edu*

Contents

1 Introduction 390

1.1 Linear parametrizations in numerical analysis 390

1.2 Nonlinear parametrizations for discretizing PDE solution fields 391

1.3 Neural Galerkin schemes 391

1.4 Literature overview 392

1.5 Outline 393

2 The need for nonlinear parametrizations in approximating solution fields of PDEs 393

2.1 Setup 393

2.2 Linear parametrizations of solution fields 394

2.3 The Kolmogorov barrier 395

2.4 Numerical illustrations of limitations of linear parametrizations 396

2.5 Nonlinear parametrizations 397

2.5.1 Generic nonlinear parametrizations 397

2.5.2 Examples of nonlinear parametrizations 398

2.5.3 Nonlinear parametrizations via time-dependent parameters 399

3 Neural Galerkin schemes based on the Dirac-Frenkel variational principle and deep networks 399

3.1 The Dirac-Frenkel variational principle 400

3.2 An optimization perspective of the Dirac-Frenkel variational principle 402

3.3 Least-squares formulation and discretization in time 402

4 Adaptive sampling in Neural Galerkin schemes 403

4.1 The sampling challenge 403

4.2 Objectives with time-dependent measures 404

4.3 A computational procedure for adaptive sampling based on particles and Stein variational gradient descent 405

4.4 Using adaptive samples in least-squares formulations of Neural Galerkin schemes 406

4.5 Example: Fokker-Planck equations in moderately high dimensions 407

5 Randomized sparse Neural Galerkin schemes 408

Handbook of Numerical Analysis, Volume 25, ISSN 1570-8659, https://doi.org/10.1016/bs.hna.2024.05.006

5.1 The importance of the tangent spaces of the parametrization manifold 409
5.2 Tangent space collapse 410
5.3 Leveraging low-rankness of batch gradients with subsampling 410
5.4 Numerical experiments with randomized sparse Neural Galerkin schemes 412
6 Conclusions 412
References 413

Abstract

This survey discusses Neural Galerkin schemes that leverage nonlinear parametrizations such as deep networks to numerically solve time-dependent partial differential equations (PDEs) in a variational sense. Neural Galerkin schemes build on the Dirac-Frenkel variational principle to train networks by minimizing the residual sequentially over time, which is in contrast to many other methods that approximate PDE solution fields with deep networks globally in time. Because of the sequential-in-time training, Neural Galerkin solutions inherently respect causality and approximate solution fields locally in time so that often fewer parameters are required than by global-in-time methods. Additionally, the sequential-in-time training enables adaptively sampling the spatial domain to efficiently evaluate the residual objectives over time, which is key for numerically realizing the expressive power of deep networks and other nonlinear parametrizations in high dimensions and when solution features are local such as wave fronts.

Keywords

Deep networks, Dirac-Frenkel variational principle, Time-dependent partial differential equations, Kolmogorov n-width, Curse of dimensionality, Model reduction

MSC Codes

65M20, 65M60, 65M22, 65F55

1 Introduction

Partial differential equations (PDEs) are broadly used in science and engineering to model systems of interest. Because analytic solutions of PDEs are available only in limited settings, one often has to resort to numerical computations to obtain approximate solutions.

1.1 Linear parametrizations in numerical analysis

A typical approach to numerically solving PDEs in numerical analysis is to first parametrize the PDE solution field with a finite number of parameters and then to derive a system of algebraic equations to solve for the parameters such that the parametrization approximates the PDE solution in some numerical sense. Common parametrizations are linear combinations of basis functions with local supports centered on grid points (Ern and Guermond, 2004). However, there are classes of PDEs for which linear parametrizations are inefficient in the sense that the best-approximation error decays slowly with increasing numbers of parameters. One important class of PDEs for which linear approximations on grid points become inefficient is given by PDEs that are formulated over high-dimensional spatial domains, which are often affected by the curse of dimensionality so that an exponential increase in the number of grid points—and thus typically computational costs—with the dimension is required to maintain the same accuracy. Another class is given by PDEs with slowly decaying Kolmogorov n-widths (Ohlberger and Rave, 2016; Greif and Urban, 2019; Arbes et al., 2023), which, e.g., provides a lower bound on the error that can be achieved with linear approximations in model reduction (Antoulas, 2005; Rozza et al., 2008; Benner et al., 2015; Antoulas et al., 2021; Kramer et al., 2024). In general, examples of PDEs that lead to slowly decaying n-widths are often found when modeling transport-dominated problems (Peherstorfer, 2022).

1.2 Nonlinear parametrizations for discretizing PDE solution fields

One approach to overcome the limitations of linear approximations is to use parametrizations that have a nonlinear dependence on the parameters. Nonlinear parametrizations are given by, for example, deep neural networks (LeCun et al., 2015), tensor networks (Orús, 2019), and Gaussian wave packets (Lubich, 2008). While nonlinear parametrizations can achieve faster error decays than linear ones under certain assumptions from an approximation-theoretic perspective, these results are mostly existence results that fall short of providing methods for numerically computing the parameters (DeVore et al., 1989, 1993; Cohen et al., 2022; DeVore et al., 2021; Daubechies et al., 2022).

1.3 Neural Galerkin schemes

The purpose of this survey is to discuss Neural Galerkin schemes (Bruna et al., 2024; Berman and Peherstorfer, 2023; Wen et al., 2024) that leverage nonlinear parametrizations to numerically solve time-dependent PDEs in a variational sense. The focus of this survey is on computational aspects rather than approximation theory. Neural Galerkin schemes build on the Dirac-Frenkel variational principle (Dirac, 1930; Frenkel, 1934; Lubich, 2008) to train networks by minimizing the residual sequentially over time, which is in contrast to many other methods that leverage deep networks that are global in time (Dissanayake and Phan-Thien, 1994; Sirignano and Spiliopoulos, 2018; Raissi et al., 2019; Berg and Nyström, 2018); see also Du and Zaki (2021); Anderson and Farazmand (2022); Kast and Hesthaven (2023); Zhang et al. (2024) for other, related sequential-in-time training methods. Because of the sequential-in-time training, Neural Galerkin solutions inherently respect causality. Furthermore, because solution fields are approximated only locally in time, often fewer parameters (e.g., networks with lower number of weights) are sufficient to provide accurate approximations when compared to global-in-time methods (Berman and Peherstorfer, 2023). In particular, locally in time, the network weights typically are of low rank, which can be leveraged via randomized sparse updates (Berman and Peherstorfer, 2023) and pretraining schemes (Berman and Peherstorfer, 2024). Additionally, the sequential-in-time training enables adaptively sampling the spatial domain to efficiently evaluate the residual objectives over time. The adaptive sampling is guided by the dynamics described by the PDE, which means that samples are placed where they are needed to efficiently evaluate the residual objective function as the solution field evolves. In fact, the numerical results in Bruna et al. (2024); Wen et al. (2024), which are also reported in this survey, show that the adaptive sampling of Neural Galerkin schemes is key for numerically realizing the expressive power of deep networks and other nonlinear parametrizations.

1.4 Literature overview

There are several other numerical methods that can leverage nonlinear parametrizations in PDE settings. Besides the large body of work on global-in-time methods such as Dissanayake and Phan-Thien (1994); Sirignano and Spiliopoulos (2018); Raissi et al. (2019); Berg and Nyström (2018), there are methods that leverage specific properties of classes of PDEs (E et al., 2017; Han et al., 2018) and focus on other, related approximation tasks such as learning committor functions (Khoo et al., 2018; Li et al., 2019; Rotskoff et al., 2022), closure modeling (Bar-Sinai et al., 2019; Kochkov et al., 2021; Wang et al., 2020), and de-noising (Rudy et al., 2019).

One major motivation for Neural Galerkin schemes is to circumvent the Kolmogorov barrier (Berman and Peherstorfer, 2024), which is a challenge in model reduction (Peherstorfer, 2022). There are other model reduction methods that can overcome the Kolmogorov barrier, and we survey several of them now. First, there are localized model reduction methods that learn a dictionary of candidate basis functions and then adaptively select a subset of basis functions (Jens et al., 2011; Dihlmann et al., 2011; Amsallem et al., 2012; Eftang and Stamm, 2012; Maday and Stamm, 2013; Peherstorfer et al., 2014; Kaulmann et al., 2015; Geelen and Willcox, 2022). By relying on a fixed dictionary, such localized model reduction methods restrict their flexibility in terms of what functions to approximate. In particular, all dynamics must have been seen in the pretraining (offline) phase. Another class of methods uses nonlinear maps to transform the solution fields such that the solution manifolds induced by the solution fields are not affected by a slowly decaying Kolmogorov n-width anymore (Rowley and Marsden, 2000; Ohlberger and Rave, 2013; Reiss et al., 2018; Ehrlacher et al., 2020; Qian et al., 2020; Papapicco et al., 2022; Issan and Kramer, 2023; Taddei et al., 2015; Cagniart et al., 2019). Related to transformations are methods based on nonlinear embeddings such as autoencoders (Lee and Carlberg, 2020; Kim et al., 2022; Romor et al., 2023). In the works (Geelen et al., 2023; Barnett and Farhat, 2022; Geelen et al., 2024; Sharma et al., 2023; Schwerdtner and Peherstorfer, 2024), approximations on quadratic manifolds have been proposed, which rely on a polynomial feature map to achieve a nonlinear parametrization.

Closer to Neural Galerkin schemes are online adaptive methods that aim to adapt basis functions over time (Koch and Lubich, 2007; Sapsis and Lermusiaux, 2009; Iollo and Lombardi, 2014; Gerbeau and Lombardi, 2014; Carlberg, 2015; Peherstorfer and Willcox, 2015; Zahr and Farhat, 2015; Peherstorfer, 2020; Black et al., 2020; Billaud-Friess and Nouy, 2017; Ramezanian et al., 2021), where a particular challenge is to achieve online updates of the basis functions efficiently (Peherstorfer and Willcox, 2015; Zimmermann et al., 2018; Peherstorfer, 2020; Huang and Duraisamy, 2023; Singh et al., 2023). Especially dynamic low-rank approximations (Koch and Lubich, 2007; Musharbash et al., 2015; Einkemmer and Lubich, 2019; Einkemmer et al., 2021; Musharbash and Nobile, 2017, 2018; Hesthaven et al., 2022) have been widely used, of which many build on the Dirac-Frenkel variational principle (Dirac, 1930; Frenkel, 1934; Lubich, 2008), just as Neural Galerkin schemes.

1.5 Outline

This manuscript is organized as follows. We first discuss the need for nonlinear parametrizations in Section 2. Neural Galerkin schemes are described in Section 3. Adaptive sampling (active data acquisition) is an important part of Neural Galerkin schemes that is discussed in Section 4. A randomized sparse extension of Neural Galerkin schemes is presented in Section 5. It numerically achieves faster runtimes and stabler approximations. Conclusions are drawn in Section 6.

2 The need for nonlinear parametrizations in approximating solution fields of PDEs

2.1 Setup

Consider a time-dependent PDE of the form

$$\partial_t q(t, \boldsymbol{x}) = f(t, \boldsymbol{x}, q), \qquad \boldsymbol{x} \in \Omega,\ t \in (0, T], \tag{1}$$

with the spatial domain $\Omega \subseteq \mathbb{R}^d$ and the solution field $q : [0, T] \times \Omega \to \mathbb{R}$. The right-hand side function f depends on time $t \in [0, T]$, the spatial coordinate $\boldsymbol{x} \in \Omega$, and the field q. The function f can depend on partial derivatives of q as well. For example, we obtain an advection-diffusion-reaction equation by setting

$$f(t, \boldsymbol{x}, q) = b(t, \boldsymbol{x}) \cdot \nabla q(t, \boldsymbol{x}) + \operatorname{div}(a(t, \boldsymbol{x}) \nabla q(t, \boldsymbol{x})) + g(t, \boldsymbol{x}, q(t, \boldsymbol{x}))$$

for coefficient functions $b : [0, \infty) \times \Omega \to \mathbb{R}^d$ and $a : [0, \infty) \times \Omega \to \mathbb{R}^d \times \mathbb{R}^d$ and a source term $g : [0, \infty) \times \Omega \times \mathbb{R} \to \mathbb{R}$. In the following, we only consider situations where the function $q(t, \cdot) : \Omega \to \mathbb{R}$ over the spatial domain Ω is in an appropriate Hilbert space $\mathcal{V}$ with inner product $\langle \cdot, \cdot \rangle_{\mathcal{V}}$ at all times $t \in [0, T]$.

We focus exclusively on Dirichlet and periodic boundary conditions, which can be imposed by restricting the space $\mathcal{V}$ so that all functions in $\mathcal{V}$ satisfy the boundary conditions. The initial conditions of (1) are denoted as $q_0 : \Omega \to \mathbb{R}$ and are elements of the set $\mathcal{V}^{(0)} \subseteq \mathcal{V}$.

Let now $\mathcal{M}$ be the set of solutions of (1), which formally is just a set of functions defined as

$$\mathcal{M} = \{q(t, \cdot) | t \in [0, T], q_0 \in \mathcal{V}^{(0)}\} \subset \mathcal{V} . \tag{2}$$

The set (2) does not necessarily have to have a manifold structure but the convention is to call it the solution manifold and we stick to this convention in the following. We also note that typically solutions of PDEs are considered in a variational sense rather than in a classical sense as in (2); however, the formulation via the classical solutions in (2) will be sufficient to demonstrate the limitations of linear parametrizations in the following.

2.2 Linear parametrizations of solution fields

Let us now consider linear parametrizations with a finite number of parameters of the solution field q, which we can write as

$$\tilde{q}(\boldsymbol{\theta}(t), \boldsymbol{x}) = \sum_{i=1}^{n} \theta_i(t) \varphi_i(\boldsymbol{x}) . \tag{3}$$

There are n parameters $\theta_1(t), \ldots, \theta_n(t) \in \mathbb{R}$, which depend on time t. We collect the parameters in the vector $\boldsymbol{\theta}(t) = [\theta_1(t), \ldots, \theta_n(t)]^T \in \mathbb{R}^n$. The functions $\varphi_1, \ldots, \varphi_n$ span a subspace of $\mathcal{V}$ of dimension at most n, in which the solution field q is approximated. Linear combinations of basis functions such as (3) are widely used in scientific computing. For example, finite-element methods (Ern and Guermond, 2004) typically discretize the spatial domain Ω with a grid and place the basis functions $\varphi_1, \ldots, \varphi_n$ at the grid points. The basis functions are often chosen with a local support in the context of finite-element methods. Another example is given by model reduction methods that build on linear combinations (3) with basis functions $\varphi_1, \ldots, \varphi_n$ that have global support (Antoulas, 2005; Rozza et al., 2008; Benner et al., 2015; Antoulas et al., 2021; Kramer et al., 2024).

The best-approximation error that can be achieved with parametrizations of the linear type (3) can be described with the concept of the Kolmogorov n-width (Pinkus, 1985). We consider the following version of the Kolmogorov n-width:

$$d_n(\mathcal{M}) = \inf_{\substack{\mathcal{V}_n \subset \mathcal{V} \\ \dim(\mathcal{V}_n) \leq n}} \sup_{q(t,\cdot) \in \mathcal{M}} \inf_{\tilde{q}^* \in \mathcal{V}_n} \|q(t, \cdot) - \tilde{q}^*\| . \tag{4}$$

The Kolmogorov n-width as given in (4) is the lowest worst-case error that any n-dimensional subspace $\mathcal{V}_n$ of $\mathcal{V}$ can achieve over the elements of $\mathcal{M}$. Note that (4) gives no indication how to construct a sequence of subspaces $(\mathcal{V}_n)_n$ that achieves $d_n(\mathcal{M})$.

The decay rate of $d_n(\mathcal{M})$ has been studied for solution manifolds induced by certain classes of PDEs. The work by Maday et al. (2002) shows an exponential decay of $d_n(\mathcal{M})$ for $\mathcal{M}$ induced by specific elliptic coercive PDEs over a single parameter. Additional results are given in Cohen and DeVore (2016). The works by Binev et al. (2011); Buffa et al. (2012); Cohen et al. (2020) show how to construct sequences of subspaces that achieve an exponential decay rate. The class of equations for which the Kolmogorov n-width decays exponentially fast are well suited for classical model reduction with linear parametrizations (Antoulas, 2005; Rozza et al., 2008; Benner et al., 2015; Antoulas et al., 2021; Kramer et al., 2024).

2.3 The Kolmogorov barrier

Let us now move from elliptic PDEs to hyperbolic ones and more generally to models that describe transport phenomena. It has been shown in Ohlberger and Rave (2016) that the linear advection equation can lead to a solution manifold that has a slowly decaying n-width. To see this, consider the equation

$$\partial_t q(t,x) + \partial_x q(t,x) = 0\,, \qquad x \in (0,1), t \in (0,1]\,, \tag{5}$$

with initial condition

$$q_0(x) = \begin{cases} 1\,, & x \leq 0 \\ 0\,, & \text{else}\,. \end{cases}$$

The solution to (5) is given by $q(t,x) = q_0(x-t)$ and thus the solution manifold $\mathcal{M}$ consists of the step functions that have a discontinuity in the spatial domain $[0,1]$. Ohlberger and Rave (2016) show the lower bound

$$d_n(\mathcal{M}) \geq c\frac{1}{\sqrt{n}}\,, \tag{6}$$

for a constant $c > 0$ independent of n. The bound (6) means that there cannot exist a sequence of subspaces $(\mathcal{V}_n)_n$ of $\mathcal{V}$ that achieves a faster error decay in the sense of (4) than $1/\sqrt{n}$, which is a slow rate compared to the exponential rate achieved for some PDEs of the elliptic type. Lower bounds with similarly slow decay rates have been shown in Arbes et al. (2023) for the linear advection equation with smoother initial conditions, where the smoothness of the initial condition determines the decay rate. A slow decay of $1/\sqrt{n}$ has also been shown for instances of the wave equation (Greif and Urban, 2019). Additionally, it is empirical observed that models that describe transport phenomena can lead to matrices of coefficient vectors with slowly decaying singular values, which is insufficient to draw conclusions about the decay of the Kolmogorov n-width but it provides indication that linear parametrizations are inefficient for such transport-dominated problems (Rowley and Marsden, 2000; Ohlberger and Rave, 2013; Reiss et al., 2018; Huang and Duraisamy, 2023; Peherstorfer, 2020; Uy et al., 2024). The slow decay of the Kolmogorov n-width is often referred to as the Kolmogorov barrier of linear parametrizations; see also the survey by Peherstorfer (2022).

2.4 Numerical illustrations of limitations of linear parametrizations

Let us consider a numerical experiment to illustrate the limitations of linear parametrizations. The following results are taken from Peherstorfer (2022). As a prototypical model of a diffusion-dominated problem, let us consider the heat equation

$$\partial_t q(t,x) - \partial_x^2 q(t,x) = 1\,, \qquad x \in \Omega\,, \tag{7}$$

with spatial domain $\Omega = (0, 1) \subset \mathbb{R}$. The boundary conditions are of homogeneous Dirichlet type and the initial condition is the zero function. We set end time to $T = 0.4$ and discretize on $N = 1024$ linear finite elements in space and with the implicit Euler method with time-step size 10^{-3} in time. Denote with $\boldsymbol{q}(t_k) \in \mathbb{R}^{1024}$ the coefficient vector of the finite-element approximation at time step $t_k = \delta t k$ for $k = 0, \dots, 400 = K$. In Fig. 1a we show the numerical solution over the time-space domain.

We now collect snapshots $\boldsymbol{q}(t_1), \dots, \boldsymbol{q}(t_K)$ over time and assemble the snapshot matrix $\boldsymbol{Q} = [\boldsymbol{q}(t_0), \boldsymbol{q}(t_1), \dots, \boldsymbol{q}(t_K)] \in \mathbb{R}^{1024\times 401}$. Recall that the decay of the singular values of the snapshot matrix $\boldsymbol{Q}$ indicates how well the snapshots can be approximated in the space spanned by the first few left-singular vectors. Fig. 1b shows the singular values, which we normalized so that the largest singular value is one. Only about 20 singular vectors are sufficient to approximate the snapshots up to double precision. A space of dimension 20 is therefore sufficient for achieving double precision even though the dimension of the snapshot vectors is $N = 1024$. It is important to note that the decay of the singular values does not allow to draw conclusions about the Kolmogorov n-width decay of the corresponding solution manifold; however, the singular values can serve as a numerical indication if classical methods with linear approximations in subspaces can be efficient.

Let us now consider a transport-dominated problem modeled by the linear advection equation (5) with spatial domain $\Omega = (0, 1)$, time domain $(0, 0.4]$ and periodic boundary conditions. We choose the Gaussian probability density function with mean 0.1 and standard deviation 1.5×10^{-2} as initial condition. The solution is plotted over time and space in Fig. 1c. Analogously to the example with the heat equation, we collect snapshot data and plot the decay of the normalized singular values in Fig. 1d. The singular values decay orders of magnitude slower, which provides further indication that solution fields of transport-dominated problems can be challenging to approximate with linear approximations in subspaces. Even though this is just a numerical illustration,

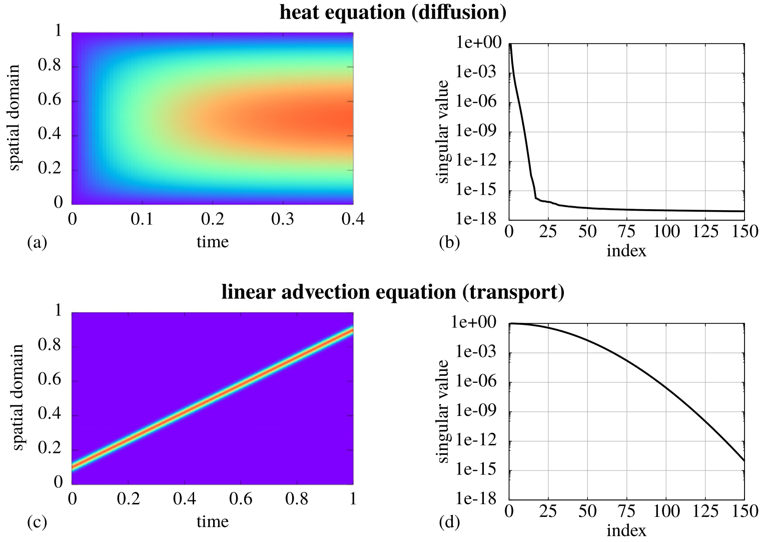


FIGURE 1 For diffusion-dominated problems such as the heat equation in this example, the singular values of snapshot matrices typically decay exponentially fast. In contrast, if dynamics are dominated by transport such as with models given by the linear advection equation, the singular values of snapshot matrices can decay orders of magnitude slower, which motivates the use of nonlinear parametrizations in model reduction for transport-dominated problems (Peherstorfer, 2022). (First published in Notices of the American Mathematical Society in 69, Number 5 (2022), published by American Mathematical Society. © 2022 American Mathematical Society.)

it is representative of the challenges of linear approximations for transport-dominated problems; see the survey by Peherstorfer (2022).

2.5 Nonlinear parametrizations

The Kolmogorov barrier can be circumvented with nonlinear parametrizations (DeVore et al., 1989, 1993; Cohen et al., 2022). We call a parametrization nonlinear if the representation can depend on the element of $\mathcal{M}$ that one wants to approximate, which is in contrast to the linear parametrizations discussed in Section 2.2, where the representation $\varphi_1, \dots, \varphi_n$ is fixed for all elements in $\mathcal{M}$.

2.5.1 Generic nonlinear parametrizations

Consider the generic nonlinear parametrization

$$\tilde{q}(\boldsymbol{\theta}(t), \boldsymbol{x}) = \sum_{i=1}^{n} \beta_i(t)\varphi_i(\boldsymbol{x}; \boldsymbol{\alpha}(t)), \tag{8}$$

with the parameter vector $\boldsymbol{\theta}(t) = [\boldsymbol{\alpha}(t); \boldsymbol{\beta}(t)] \in \mathbb{R}^p$ that consists of the feature vector $\boldsymbol{\alpha}(t) = [\alpha_1(t), \dots, \alpha_{n'}(t)]^T$ with n' components and the vector

$\boldsymbol{\beta}(t) = [\beta_1(t), \dots, \beta_n(t)]^T$ of n coefficients that enter $\tilde{q}$ linearly. The representation given by the functions $\varphi_1(\cdot; \boldsymbol{\alpha}(t)), \dots, \varphi_n(\cdot; \boldsymbol{\alpha}(t))$ depends on the time-dependent feature vector $\boldsymbol{\alpha}(t)$ and thus can change over time t based on the element $q(t, \cdot) \in \mathcal{M}$ of the solution manifold. In other words, the representation $\varphi_1, \dots, \varphi_n$ can be adapted based on the element $q(t, \cdot)$ that is to be approximated and thus (8) is a nonlinear parametrization. In this sense, adaptive mesh refinement in scientific computing, which was introduced for hyperbolic problems in Berger and Colella (1989); Berger and LeVeque (1998), is also a form of nonlinear parametrization because the basis functions are adapted based on how the solution fields evolve over time.

2.5.2 *Examples of nonlinear parametrizations*

The nonlinear parametrization given in (8) is generic and we now consider specific instances of it. Let us first consider dictionary approaches, which we can write as (8) by restricting us to have only a finite number $L \in \mathbb{N}$ of feature vectors $\boldsymbol{\alpha}^{(1)}, \dots, \boldsymbol{\alpha}^{(L)} \in \mathbb{R}^{n'}$ that are independent of time t. Each feature vector $\boldsymbol{\alpha}^{(i)}$ corresponds to a representation $\varphi_1^{(i)} = \varphi_1(\cdot; \boldsymbol{\alpha}^{(i)}), \dots, \varphi_n^{(i)} = \varphi_n(\cdot; \boldsymbol{\alpha}^{(i)})$ and a corresponding subspace $\mathcal{V}_n^{(i)} \subset \mathcal{V}$ for $i = 1, \dots, L$. Based on a classification function such as $I : \mathcal{V} \to \{1, \dots, L\}$, one of the representations is selected based on the element of $\mathcal{M}$ that is to be approximated. Because the representation is chosen based on the element that is to be approximated, it provides a nonlinear parametrization. Such nonlinear parametrizations with a finite number of feature vectors have been studied extensively in the context of model reduction under the umbrella term of localized model reduction (Jens et al., 2011; Dihlmann et al., 2011; Amsallem et al., 2012; Eftang and Stamm, 2012; Maday and Stamm, 2013; Peherstorfer et al., 2014; Kaulmann et al., 2015; Geelen and Willcox, 2022). Localized model reduction is closely related to dictionary approaches, because the combination of all functions $\varphi_1^{(1)}, \dots, \varphi_n^{(1)}, \varphi_1^{(2)}, \dots, \varphi_n^{(L)}$ can be considered as the dictionary from which an indicator function selects n elements.

Another type of nonlinear parametrizations that is widely used in model reduction is building on nonlinear transformations. For example, if the manifold $\mathcal{M}$ describes solution fields with moving coherent structures in the spatial domain, then the functions $\varphi_1, \dots, \varphi_n$ in (8) can account for this transport via the feature vector $\boldsymbol{\alpha}(t)$. A frequently given example is the linear advection equation (5), which has as solution $q(t, x) = q_0(x - t)$ and thus setting $n = 1$ and $n' = 1$ with $\alpha_1(t) = -t$ so that $\varphi_1(x, \boldsymbol{\alpha}(t)) = q_0(x - t)$ provides an exact representation of the solution field; see also Peherstorfer (2022). More sophisticated transformations can be constructed either analytically (Rowley et al., 2004; Ehrlacher et al., 2020) or via optimization (Reiss et al., 2018; Taddei, 2020; Taddei and Zhang, 2021).

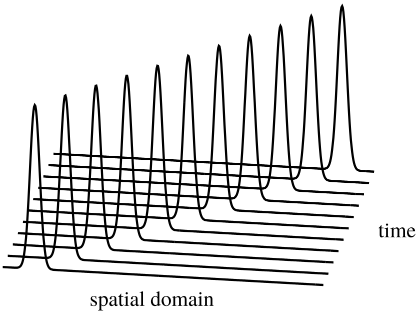


FIGURE 2 The classical method of lines first discretizes the spatial domain of time-dependent partial differential equations to obtain a system of ODEs, which is then integrated in time to obtain approximate coefficient vectors for linear combinations of basis functions that represent the solution field. Neural Galerkin schemes are also sequentially in time approximating solution fields but allow to use nonlinear parametrizations such as deep networks.

2.5.3 Nonlinear parametrizations via time-dependent parameters

In this survey, we consider nonlinear parametrizations with time-dependent parameters $\boldsymbol{\theta}(t)$. In the following, there is no need to distinguish between features $\boldsymbol{\alpha}(t)$ and coefficients $\boldsymbol{\beta}(t)$ and thus we can consider the nonlinear parametrization as just depending on a p-dimensional vector $\boldsymbol{\theta}(t)$ as

$$\tilde{q}(\boldsymbol{\theta}(t), \cdot) : \Omega \to \mathbb{R} . \tag{9}$$

Examples of such parametrizations are given by dynamic low-rank approximations (Koch and Lubich, 2007; Sapsis and Lermusiaux, 2009), deep neural networks with time-dependent weights (Bruna et al., 2024; Du and Zaki, 2021; Finzi et al., 2023), tensor networks (Orús, 2019), Gaussian wave packets (Lubich, 2008), and other nonlinear parametrizations (Black et al., 2020).

3 Neural Galerkin schemes based on the Dirac-Frenkel variational principle and deep networks

We now discuss Neural Galerkin schemes (Bruna et al., 2024) that provide dynamical systems for the time-dependent parameters $\boldsymbol{\theta}(t)$ so that the nonlinear parametrizations $\tilde{q}(\boldsymbol{\theta}(t), \cdot)$ solve the PDEs of interest in a variational sense; see Fig. 2. The two key building blocks of Neural Galerkin schemes are the Dirac-Frenkel variational principle (Lasser and Lubich, 2020, Section 3.8) for deriving the dynamical systems and deep networks with time-dependent weights for providing the nonlinear parametrizations.

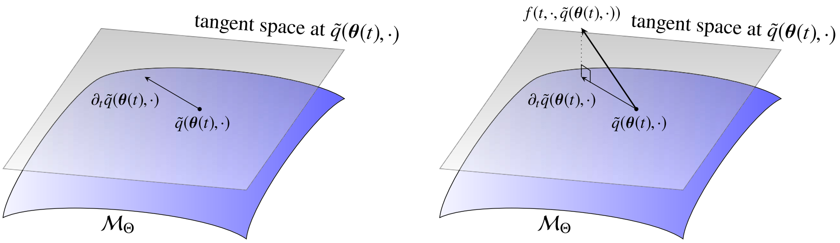

tangent space at $\tilde{q}(\boldsymbol{\theta}(t),\cdot)$
$\partial_t\tilde{q}(\boldsymbol{\theta}(t),\cdot)$
$\tilde{q}(\boldsymbol{\theta}(t),\cdot)$
$\mathcal{M}_\Theta$
$f(t,\cdot,\tilde{q}(\boldsymbol{\theta}(t),\cdot))$
tangent space at $\tilde{q}(\boldsymbol{\theta}(t),\cdot)$
$\partial_t\tilde{q}(\boldsymbol{\theta}(t),\cdot)$
$\tilde{q}(\boldsymbol{\theta}(t),\cdot)$
$\mathcal{M}_\Theta$

FIGURE 3 The time derivative $\dot{\boldsymbol{\theta}}(t)$ of the parameter $\boldsymbol{\theta}(t)$ is given by approximating the right-hand side function $f(t,\cdot,\tilde{q}(\boldsymbol{\theta}(t),\cdot))$ in the tangent space of the parametrization manifold $\mathcal{M}_\Theta$ at the current solution $\tilde{q}(\boldsymbol{\theta}(t),\cdot)$ in a least-squares sense, which is the Dirac-Frenkel variational principle (Lasser and Lubich, 2020, Section 3.8).

3.1 The Dirac-Frenkel variational principle

Recall the nonlinear parametrizations with p parameters described in (9). Let us drop the time dependence of $\boldsymbol{\theta}$ for now to obtain

$$\tilde{q}(\boldsymbol{\theta},\cdot):\Omega\to\mathbb{R}, \tag{10}$$

with $\boldsymbol{\theta}\in\Theta\subseteq\mathbb{R}^p$. The parametrization (10) induces the manifold

$$\mathcal{M}_\Theta=\{\tilde{q}(\boldsymbol{\theta},\cdot)\,|\,\boldsymbol{\theta}\in\Theta\}\ ,$$

which is different from the manifold $\mathcal{M}$ induced by the PDE solutions; see Section 2.1. The manifold $\mathcal{M}_\Theta$ is depicted in Fig. 3a. The tangent space of $\mathcal{M}_\Theta$ at a point $\tilde{q}(\boldsymbol{\theta},\cdot)\in\mathcal{M}_\Theta$ is spanned by the component functions of the gradient $\nabla_{\boldsymbol{\theta}}\tilde{q}$,

$$\mathcal{T}_{\tilde{q}(\boldsymbol{\theta},\cdot)}\mathcal{M}_\Theta=\operatorname{span}\left\{\partial_{\theta_1}\tilde{q},\ldots,\partial_{\theta_p}\tilde{q}\right\}\,. \tag{11}$$

Let us now make the parameter $\boldsymbol{\theta}(t)$ depend on time t again so that we can consider the time derivative of $\tilde{q}(\boldsymbol{\theta}(t),\cdot)$. Notice that time t enters the parametrization $\tilde{q}$ only via the parameter vector $\boldsymbol{\theta}(t)$. We thus apply the chain rule to obtain

$$\partial_t\tilde{q}(\boldsymbol{\theta}(t),\cdot)=\nabla_{\boldsymbol{\theta}}\tilde{q}(\boldsymbol{\theta}(t),\cdot)\cdot\dot{\boldsymbol{\theta}}(t)\,, \tag{12}$$

where $\dot{\boldsymbol{\theta}}(t)$ is a vector in $\mathbb{R}^p$, which we interpret as the time derivative of the parameter vector $\boldsymbol{\theta}(t)$. Eq. (12) shows that the time derivative $\partial_t\tilde{q}(\boldsymbol{\theta}(t),\cdot)$ is an element of the tangent space $\mathcal{T}_{\tilde{q}(\boldsymbol{\theta}(t),\cdot)}\mathcal{M}_\Theta$ of $\mathcal{M}_\Theta$ at $\tilde{q}(\boldsymbol{\theta}(t),\cdot)$, which is spanned by the component functions of the gradient $\nabla_{\boldsymbol{\theta}}\tilde{q}(\boldsymbol{\theta}(t),\cdot)$.

Let us now consider the PDE given in (1) again so that we can define the residual function at time t as

$$r_t(\boldsymbol{\theta}(t),\dot{\boldsymbol{\theta}}(t),\cdot)=\underbrace{\partial_t\tilde{q}(\boldsymbol{\theta}(t),\cdot)}_{\nabla_{\boldsymbol{\theta}}\tilde{q}(\boldsymbol{\theta}(t),\cdot)\cdot\dot{\boldsymbol{\theta}}(t)}-f(t,\cdot,\tilde{q}(\boldsymbol{\theta}(t),\cdot))\,, \tag{13}$$

where we plugged $\tilde{q}(\boldsymbol{\theta}(t), \cdot)$ into (1). Notice that the residual function r_t has the parameter vector $\boldsymbol{\theta}(t)$ as well as the time derivative $\dot{\boldsymbol{\theta}}(t)$ as arguments. We now discuss multiple options to seek $\dot{\boldsymbol{\theta}}(t)$, which all will result in the same dynamics and thus be equivalent. One option is to project the right-hand side function f onto the tangent space and then to seek a $\dot{\boldsymbol{\theta}}(t)$ that sets the corresponding residual to zero. Because we know that the time derivative $\partial_t \tilde{q}(\boldsymbol{\theta}(t), \cdot)$ is in the tangent space (11), such a $\dot{\boldsymbol{\theta}}(t)$ can be found, which closes the equation,

$$\partial_t \tilde{q}(\boldsymbol{\theta}(t), \cdot) = \mathrm{P}_{\boldsymbol{\theta}(t)} f(t, \cdot, \tilde{q}(\boldsymbol{\theta}(t), \cdot)) \tag{14}$$

with the projection with respect to an L^2 inner product $\langle \cdot, \cdot \rangle_\nu$ with measure ν

$$\mathrm{P}_{\boldsymbol{\theta}} g = \nabla_{\boldsymbol{\theta}} \tilde{q} \cdot \langle \nabla_{\boldsymbol{\theta}} \tilde{q}, g \rangle_\nu ,$$

for $g \in \mathcal{V}$. Eq. (14) is found in a wide range of literature that builds on the Dirac-Frenkel variational principle (Dirac, 1930; Frenkel, 1934; Lubich, 2008; Koch and Lubich, 2007; Sapsis and Lermusiaux, 2009; Hesthaven et al., 2022; Du and Zaki, 2021; Anderson and Farazmand, 2022). Another option, that will result in the same equation (14), can be derived by imposing Galerkin conditions for finding $\dot{\boldsymbol{\theta}}(t)$ so that the residual r_t is orthogonal to the tangent space with respect to the inner product $\langle \cdot, \cdot \rangle_\nu$,

$$\langle \partial_{\theta_i} \tilde{q}(\boldsymbol{\theta}(t), \cdot), r_t(\boldsymbol{\theta}(t), \dot{\boldsymbol{\theta}}(t), \cdot) \rangle_\nu = 0 , \quad i = 1, \ldots, p . \tag{15}$$

Yet another option is to minimize the squared residual norm locally over time, which once more will lead to the same equation dynamics; see Section 3.2.

Transformations show that (14) and equivalently (15) provide the dynamics of the parameter vector $\boldsymbol{\theta}(t)$ as

$$M(t, \boldsymbol{\theta}(t)) \dot{\boldsymbol{\theta}}(t) = F(t, \boldsymbol{\theta}(t)) , \qquad \boldsymbol{\theta}(0) = \boldsymbol{\theta}^{(0)} , \tag{16}$$

with

$$\begin{aligned} M(t, \boldsymbol{\theta}) &= \int_\Omega \nabla_{\boldsymbol{\theta}} \tilde{q}(\boldsymbol{\theta}, \boldsymbol{x}) \otimes \nabla_{\boldsymbol{\theta}} \tilde{q}(\boldsymbol{\theta}, \boldsymbol{x}) \mathrm{d}\nu(\boldsymbol{x}) , \\ F(t, \boldsymbol{\theta}) &= \int_\Omega \nabla_{\boldsymbol{\theta}} \tilde{q}(\boldsymbol{\theta}, \boldsymbol{x}) f(t, \boldsymbol{x}, \tilde{q}(\boldsymbol{\theta}, \cdot)) \mathrm{d}\nu(\boldsymbol{x}) , \end{aligned}$$

and an initial condition $\boldsymbol{\theta}^{(0)} \in \Theta$. The symbol $\otimes$ means the outer product of the gradients. We refer to (16) as the Neural Galerkin equations because they use deep networks as a parametrization and correspond to the Galerkin conditions (15). But we stress that analogous equations have been derived for various other parametrizations as well as deep networks under various names (Lubich, 2008; Koch and Lubich, 2007; Sapsis and Lermusiaux, 2009; Hesthaven et al., 2022; Du and Zaki, 2021; Anderson and Farazmand, 2022). Because the matrix $M(t, \boldsymbol{\theta})$ in (16) can become singular, the dynamical system can have multiple solutions $\boldsymbol{\theta}(t)$; see Section 5 for a more detailed discussion.

3.2 An optimization perspective of the Dirac-Frenkel variational principle

Many methods in machine learning are motivated via an optimization perspective rather than the Galerkin projection perspective often found in scientific computing. Let us now derive the dynamical system given by the Dirac-Frenkel variational principle by starting with an objective function

$$H_t(\boldsymbol{\theta}, \eta) = \mathbb{E}_{\boldsymbol{x} \sim \nu}\left[|r_t(\boldsymbol{\theta}, \eta, \boldsymbol{x})|^2 \right], \tag{17}$$

as in Du and Zaki (2021). We then seek $\dot{\boldsymbol{\theta}}(t)$ that minimizes objective H_t at time t with respect to η,

$$\min_{\eta \in \Theta} H_t(\boldsymbol{\theta}(t), \eta)$$

Problem (17) does not necessarily have a unique global optimum. We only ask for first-order optimality so that it is sufficient to set the gradient of H_t with respect to η to zero,

$$\nabla_\eta J_t(\boldsymbol{\theta}(t), \dot{\boldsymbol{\theta}}(t)) = 0, \tag{18}$$

which leads to the dynamical system for $\boldsymbol{\theta}(t)$ given in (16).

A remark is in order. The optimization perspective given by introducing the dynamical system (16) via the objective function (17) clearly contrasts the present approach to widely used time-space discretizations, which parametrize the whole time-space domain with a neural network (Raissi et al., 2019). In this sense, Neural Galerkin schemes can be interpreted as being nonlinear extensions of the classical method of lines (Zafarullah, 1970; Verwer and Sanz-Serna, 1984) for solving time-dependent PDEs, where first the spatial domain is discretized to obtain a semidiscrete system of ordinary differential equations (ODEs). The system of ODEs is then discretized in time and numerically integrated. Similarly, Neural Galerkin schemes first parametrize the spatial domain of the PDEs with time-dependent parametrizations, which are then numerically computed by integrating a dynamical system in time. Again, this is in contrast to time-space approximations with deep networks as Raissi et al. (2019).

3.3 Least-squares formulation and discretization in time

The dynamical system (16) for $\boldsymbol{\theta}(t)$ describes the normal equations of the least-squares problem

$$\min_{\dot{\boldsymbol{\theta}}(t) \in \Theta} \|\nabla_{\boldsymbol{\theta}} \tilde{q}(\boldsymbol{\theta}(t), \cdot) \cdot \dot{\boldsymbol{\theta}}(t) - f(t, \cdot, \tilde{q}(\boldsymbol{\theta}(t), \cdot))\|_\nu . \tag{19}$$

Numerically it thus is often beneficial with respect to the condition of the problem to work with the least-squares problem (19) rather than the dynamical system (16). An important insight is that the unknown $\dot{\boldsymbol{\theta}}(t)$ enters linearly in the least-squares problem (19), even though the parametrization $\tilde{q}$ depends nonlinearly on the parameter vector $\boldsymbol{\theta}(t)$.

We can now discretize (19) in time. Consider therefore $K \in \mathbb{N}$ time steps with $0 = t_0 < t_1 < \cdots < t_K = T$. At each time step $k = 0, \ldots, K-1$, we obtain a parameter vector $\boldsymbol{\theta}_k \in \mathbb{R}^p$, which is an approximation of the time-continuous parameter $\boldsymbol{\theta}(t_k)$ at time step t_k. If we take an explicit time integration scheme such as forward Euler, we obtain the least-squares problems

$$\min_{\boldsymbol{\theta}_{k+1} \in \Theta} \|\nabla_{\boldsymbol{\theta}} \tilde{q}(\boldsymbol{\theta}_k, \cdot) \cdot (\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k) - \delta t f(t_k, \cdot, \tilde{q}(\boldsymbol{\theta}_k, \cdot))\|_\nu \,, k = 0, \ldots, K-1\,. \tag{20}$$

Because we used an explicit time integration scheme, the linearity of the time-continuous least-squares problem (19) is preserved in (20): At each time step $k = 0, \ldots, K-1$, a linear least-squares problem has to be solved. In contrast, if we take an implicit time integration scheme such as backward Euler, then we obtain

$$\begin{aligned} &\min_{\boldsymbol{\theta}_{k+1} \in \Theta} \|\nabla_{\boldsymbol{\theta}} \tilde{q}(\boldsymbol{\theta}_{k+1}, \cdot) \cdot (\boldsymbol{\theta}_{k+1} - \boldsymbol{\theta}_k) - \delta t f(t_{k+1}, \cdot, \tilde{q}(\boldsymbol{\theta}_{k+1}, \cdot))\|_\nu \,, \\ &\qquad k = 0, \ldots, K-1\,, \end{aligned} \tag{21}$$

where now the unknown $\boldsymbol{\theta}_{k+1}$ enters nonlinearly via the parametrizations $\tilde{q}$. Thus, at each time step, a nonlinear optimization problem has to be solved, which can be computationally more expensive than solving the linear least-squares problems corresponding to the explicit schemes. The observation that explicit time integration schemes lead to computationally cheaper time steps than implicit ones is often the case in numerical analysis and scientific computing and reflects that Neural Galerkin schemes are rooted in numerical analysis. As a side remark, we state that constraints can be added to (19) and their discrete counterparts to conserve mass, momentum, Hamiltonians (Schwerdtner et al., 2023); see also Anderson and Farazmand (2022) for a method based on nonlinear parametrizations with conserving quantities.

4 Adaptive sampling in Neural Galerkin schemes

To numerically solve for the parameter $\boldsymbol{\theta}(t)$ in Neural Galerkin schemes, the objective of the least-squares problem (19) has to be numerically evaluated, which is the topic of this section.

4.1 The sampling challenge

The least-squares problem (19) and its discrete counterparts (20) and (21) are formulated with objectives that depend on the norm $\|\cdot\|_\nu$ with measure ν over the spatial domain Ω. To numerically optimize for the parameter $\boldsymbol{\theta}(t)$, it is necessary to numerically estimate the objectives and thus to evaluate the norm $\|\cdot\|_\nu$ via the inner product $\langle \cdot, \cdot \rangle_\nu$. If we can draw samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m \sim \nu$ from the distribution corresponding to the measure ν, then we can estimate the objective of (19) via a Monte Carlo estimator as

$$\frac{1}{m}\sum_{i=1}^{m} |\nabla_{\boldsymbol{\theta}} \tilde{q}(\boldsymbol{\theta}(t), \boldsymbol{x}_i) \cdot \dot{\boldsymbol{\theta}}(t) - f(t, \boldsymbol{x}_i, \tilde{q}(\boldsymbol{\theta}(t), \cdot))|^2 . \tag{22}$$

Efficiently evaluating objective functions with Monte Carlo estimators such as (22) is a pervasive challenge in machine learning, where it is typically referred to as estimating the population loss with the empirical loss (Vapnik, 1991).

Analogously, in scientific computing, inner products have to be evaluated in, e.g., finite-element methods to assemble systems of equations corresponding to Galerkin conditions. Linear parametrizations such as linear combinations of basis functions centered on grid points allow in many cases to precompute inner products of local basis elements, which then can be re-used to efficiently evaluate inner products globally (Ern and Guermond, 2004). In case of nonlinear parametrizations, the superposition principle of linear approximations is lost and thus the inner product needs to be evaluated explicitly at each time step. The evaluation of the objective can be a major numerical runtime bottleneck when working with nonlinear parametrizations (Wen et al., 2024).

In certain limited cases, objectives can be evaluated analytically (Lubich, 2008; Lasser and Lubich, 2020). In other settings, it has been proposed to build on quadrature rules (Du and Zaki, 2021; Schwerdtner et al., 2023), which can be sufficient for problems with low-dimensional spatial domains. But even in low dimensions, local features such as wave fronts can make nonadaptive quadrature rules inefficient. Another common option therefore is falling back to Monte Carlo estimators that simply sample from the measure ν in the spatial domain to estimate the objectives. However, in particular for transport-dominated problems where local features such as wave fronts move through the spatial domain, a static sampling from a measure that is fixed over all times t means that these local features that change over time t need to be discovered without guidance; see Rotskoff et al. (2022); Bruna et al. (2024); Wen et al. (2024). For example, consider the linear advection equation with a Gaussian bump as initial condition, for which we plotted the solution field over the time-space domain in Fig. 1. A uniform sampling of the domain quickly reveals that many samples are required to resolve the local Gaussian bump as it is transported through the spatial domain. Thus, even though the nonlinear parametrization can be expressive, evaluating the objective function to numerically find parameters that realize the expressiveness can still be challenging.

4.2 Objectives with time-dependent measures

The works by Bruna et al. (2024); Wen et al. (2024) propose to consider time-dependent inner products $\langle \cdot, \cdot \rangle_{\mu_t}$ for formulating the least-squares problem (19) to obtain

$$\min_{\dot{\boldsymbol{\theta}}(t)\in\Theta} \|\nabla_{\boldsymbol{\theta}}\tilde{q}(\boldsymbol{\theta}(t),\cdot)\cdot\dot{\boldsymbol{\theta}}(t) - f(t,\cdot,\tilde{q}(\boldsymbol{\theta}(t),\cdot))\|_{\mu_t} \tag{23}$$

in continuous time. The inner product depends on time via the time-dependent measure μ_t, which is in contrast to the formulation (19) that builds on a measure ν that is fixed in time. As shown in Wen et al. (2024), if the parametrization is so rich that the residual is zero, then the optima of the objective of (19) are invariant to the measure as long as the measure has full support in the spatial domain. If a zero residual is not reached, then this insight serves as a heuristic when the norm of residual is small. Notice that we are indeed interested in cases where the norm of the residual is small because otherwise it would mean we incur large errors during time integration. Thus, in this sense, we are free to choose measure μ_t as long as it is fully supported on the spatial domain Ω.

The goal is now to choose a measure μ_t such that the objective of (23) can be estimated accurately with few samples. We set $\mu_t \propto \exp(-V_{\boldsymbol{\theta}(t),\dot{\boldsymbol{\theta}}(t)})$ with the potential

$$V_{\boldsymbol{\theta}(t),\dot{\boldsymbol{\theta}}(t)}(\boldsymbol{x}) = |r_t(\boldsymbol{\theta}(t),\dot{\boldsymbol{\theta}}(t),\boldsymbol{x})|^2\,, \tag{24}$$

where r_t is the residual function defined in (13). The measure μ_t defined via (24) distributes mass proportional to the magnitude of the residual, which agrees with the intuition that one should sample where the residual is high.

The time-dependent measure μ_t is then coupled to the dynamics of the parameter vector $\boldsymbol{\theta}(t)$ because the potential $V_{\boldsymbol{\theta}(t),\dot{\boldsymbol{\theta}}(t)}$ of μ_t depends on $\boldsymbol{\theta}(t)$ and vice versa. Thus, the parameter vector $\boldsymbol{\theta}(t)$ and the measure μ_t are updated together over time via the coupled dynamical system

$$\begin{aligned} M(t,\boldsymbol{\theta}(t))\dot{\boldsymbol{\theta}}(t) =& F(t,\boldsymbol{\theta}(t))\,,\\ \partial_t\mu_t =& \gamma\nabla\cdot(\nabla\mu_t + \mu_t\nabla V_{\boldsymbol{\theta}(t),\dot{\boldsymbol{\theta}}(t)})\,, \end{aligned} \tag{25}$$

where γ is a scaling parameter that controls how much faster the measure μ_t moves forward in time versus the parameter vector $\boldsymbol{\theta}(t)$; details can be found in Wen et al. (2024).

Note that other options than sampling proportional to the squared residual are possible such as sampling proportional to the magnitude of the solution field $\tilde{q}$, which can be useful if the solution field itself is a density as in the case when the PDE (1) is a Fokker-Planck equation.

4.3 A computational procedure for adaptive sampling based on particles and Stein variational gradient descent

Let us discretize the time-dependent measure μ_t introduced in the previous section with an empirical measure $\tilde{\mu}_t$ that depends on a set of particles $\{\boldsymbol{x}_i(t)\}_{i=1}^m$

at time t

$$\tilde{\mu}_t = \frac{1}{m} \sum\nolimits_{i=1}^{m} \delta_{\boldsymbol{x}_i(t)} \,.$$

We can evaluate gradients of the potential V, which means we can evaluate the score of the density of μ_t,

$$\nabla \log \mu_t \,.$$

Building on Stein variational gradient descent (SVGD) with a kernel $\mathcal{K}$ (Liu and Wang, 2016), we obtain a system of ODEs for the particles,

$$\frac{\mathrm{d}}{\mathrm{d}\tau} \boldsymbol{x}_i^{(\tau)}(t_k) = \mathbb{E}_{\boldsymbol{x}' \sim \tilde{\mu}_{t_k}^{(\tau)}} \left[\mathcal{K}(\boldsymbol{x}', \boldsymbol{x}_i^{(\tau)}(t_k)) \nabla \log \mu_{t_k}(\boldsymbol{x}') + \nabla_1 \mathcal{K}(\boldsymbol{x}', \boldsymbol{x}_i^{(\tau)}(t_k)) \right] ,$$

where τ is an artificial time in which the particles move, which is different from the physical time t. The relation between the artificial time τ and the physical time t is controlled by the parameter γ in the dynamics (25); see Wen et al. (2024) for details. Other sampling techniques than SVGD can be used such as Langevin and Markov chain Monte Carlo methods. In discrete time, it is beneficial to initialize the empirical measure at time t_{k+1} with the empirical measure from time t_k and thus particle methods are particularly well suited here.

4.4 Using adaptive samples in least-squares formulations of Neural Galerkin schemes

Recall the time-dependent sampling points, which we called particles in the previous section,

$$\boldsymbol{x}_1(t), \dots, \boldsymbol{x}_m(t) \sim \mu_t \,.$$

We use the samples to form the batch gradient $\boldsymbol{J}_t(\boldsymbol{\theta}(t)) \in \mathbb{R}^{m \times p}$ as

$$\boldsymbol{J}_t(\boldsymbol{\theta}(t)) = \begin{bmatrix} - & \nabla_{\boldsymbol{\theta}} \tilde{q}(\boldsymbol{\theta}(t), \boldsymbol{x}_1(t))^T & - \\ & \vdots & \\ - & \nabla_{\boldsymbol{\theta}} \tilde{q}(\boldsymbol{\theta}(t), \boldsymbol{x}_m(t))^T & - \end{bmatrix} , \tag{26}$$

and the batch right-hand side $\boldsymbol{f}_t(\boldsymbol{\theta}(t)) \in \mathbb{R}^m$ as

$$\boldsymbol{f}_t(\boldsymbol{\theta}(t)) = \begin{bmatrix} f(t, \boldsymbol{x}_1(t), \tilde{q}(\boldsymbol{\theta}(t), \cdot)) \\ \vdots \\ f(t, \boldsymbol{x}_m(t), \tilde{q}(\boldsymbol{\theta}(t), \cdot)) \end{bmatrix} . \tag{27}$$

If we now discretize with the explicit Euler method in time, we obtain the regression problems

$$\min_{\delta\boldsymbol{\theta}_k \in \Theta} \| \hat{\boldsymbol{J}}_{t_k}(\boldsymbol{\theta}_k) \delta\boldsymbol{\theta}_k - \hat{\boldsymbol{f}}_{t_k}(\boldsymbol{\theta}_k) \|_2^2$$

over time steps $k = 0, \ldots, K-1$, where $\hat{\boldsymbol{J}}_{t_k}$ and $\hat{\boldsymbol{f}}_{t_k}$ are the time-discrete counterparts of $\boldsymbol{J}_t$ and $\boldsymbol{f}_t$, respectively. At each time step $k = 0, \ldots, K-1$, we update $\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \delta t_k \delta\boldsymbol{\theta}_k$. Notice that the time-step size $\delta t_k > 0$ can depend on the time step, which allows adaptively choosing the time-step size. We alternate between taking a time step to compute $\boldsymbol{\theta}_{k+1}$ from $\boldsymbol{\theta}_k$ and updating the set of particles from $\{\boldsymbol{x}(t_k)\}_{i=1}^m$ to $\{\boldsymbol{x}(t_{k+1})\}_{i=1}^m$; see Wen et al. (2024) for details.

4.5 Example: Fokker-Planck equations in moderately high dimensions

We report the experiment of Bruna et al. (2024) here. Consider a system of $d = 8$ particles that are attracted by a time-varying trap. The positions $X_1(t) \ldots, X_d(t) \in \mathbb{R}$ of the particles are governed by the stochastic differential equation

$$\mathrm{d}X_i = -(X_i - a(t))\mathrm{d}t - \frac{\alpha}{d}\sum\nolimits_{j=1}^{d}(X_i - X_j)\mathrm{d}t + \sqrt{2\beta^{-1}}\mathrm{d}W_i\,, \quad i = 1, \ldots, d\,,$$

with the Wiener processes W_i, $\alpha = 1/4$, $\beta = 10^2$, and transport coefficient

$$a(t) = 5/4(\sin(\pi t) + 3/2)\,.$$

The density of the particle positions is governed by the Fokker-Planck equation over the $d = 8$ dimensional spatial domain. As the particles get trapped, the density concentrates, which leads to local features in high dimensions. We solve for the density function with Neural Galerkin schemes, with a shallow network with Gaussian units and 30 nodes per layer. Time is discretized with Runge-Kutta 4 and time-step size $\delta t = 10^{-3}$. The adaptive sampling is based on $m = 1000$ particles. The particles are adapted by sampling proportional to the magnitude of the current solution function. Details of the numerical setup are provided in Bruna et al. (2024).

Fig. 4 shows the positions of particles X_1, X_4, X_8 as well as of particles X_6, X_7, X_8. A benchmark solution is computed, which is indicated via black dots. The Neural Galerkin solution with adaptive sampling closely matches the benchmark, whereas the static sampling over the spatial domain leads too poor approximations of the particle positions. A quantitive comparison is shown in Fig. 5. The relative error in the mean particle position is on the order of 10^{-3} with adaptive sampling, whereas a static sampling leads to a relative error larger than one. The covariance of the particle distribution is approximated to a relative error of about 10^{-2} in this example.

Because we compute the density function, we can compute quantities of interest that require more than just the moments, as provided by Monte Carlo methods. We thus can compute the entropy of the system, which we show in Fig. 6a. Comparing to a Monte Carlo sampling with subsequent density estimation shows that the Neural Galerkin approximation leads to more accurate

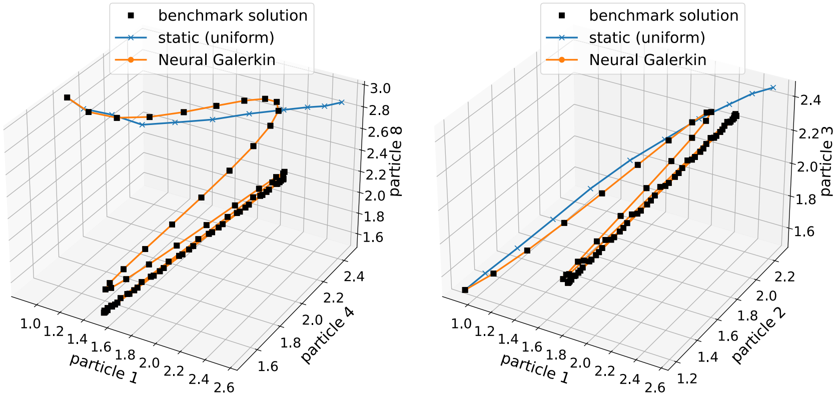
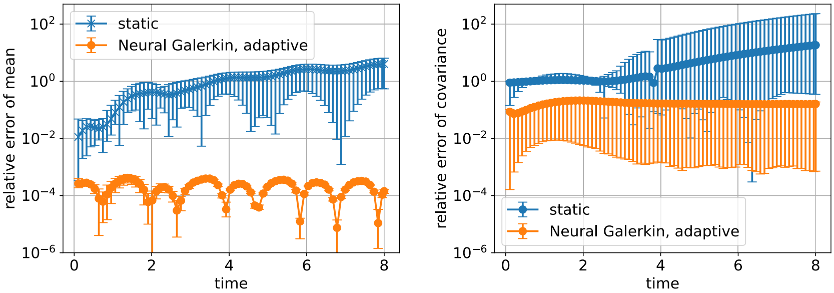


FIGURE 4 Neural Galerkin schemes with adaptive sampling accurately predict the positions of the physics particles, whereas using the same network as in the Neural Galerkin solution but with a static, uniform sampling fails to provide accurate predictions of the particle positions. (Figure from Bruna et al. (2024).)



(a) mean of particle density

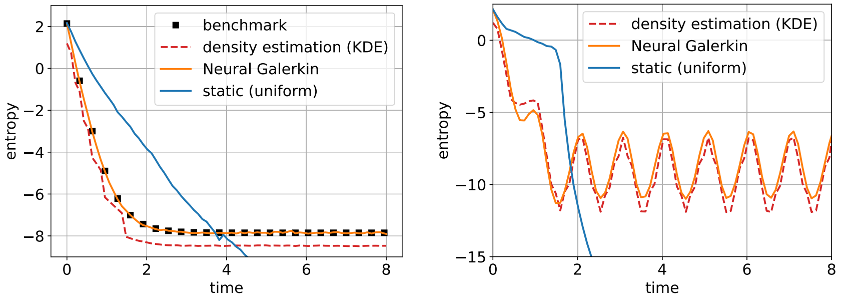
(b) covariance of particle density

FIGURE 5 Neural Galerkin schemes with adaptive sampling achieve accurate approximations of the mean and covariance of the particle density, whereas a static, uniform sampling leads to large relative errors. (Figure from Bruna et al., 2024.)

entropy estimates. We also show in Fig. 6b the entropy computed for an aharmonic trap; details in Bruna et al. (2024). It is known that such a system has an oscillating entropy, which agrees with the prediction of Neural Galerkin in this case.

5 Randomized sparse Neural Galerkin schemes

The work by Berman and Peherstorfer (2023) introduces randomized sparse Neural Galerkin (RSNG) schemes that update only sparse subsets of the components of $\boldsymbol{\theta}(t)$ and randomize which components of $\boldsymbol{\theta}(t)$ are updated. Updating only a sparse subset is sufficient because many nonlinear parametrizations lead to batch gradients $\boldsymbol{J}_t(\boldsymbol{\theta})$ that are of low rank, which indicates that components of



(a) entropy, harmonic trap

(b) entropy, aharmonic trap

FIGURE 6 With Neural Galerkin schemes in this example, we approximate the particle density function rather than only moments. We thus can compute quantities such as the entropy. In contrast, to estimate the entropy with Monte Carlo-based methods, the density function has to be estimated, which is challenging and leads to less accurate results in this example than the Neural Galerkin schemes. (Figure from Bruna et al., 2024.)

the time derivative $\dot{\boldsymbol{\theta}}(t)$ are redundant and can be ignored for updating $\boldsymbol{\theta}(t)$ without losing expressiveness. Additionally, the randomization can be interpreted analogously to dropout (Srivastava et al., 2014; Sung et al., 2021; Zaken et al., 2022), which helps preventing overfitting locally in time.

5.1 The importance of the tangent spaces of the parametrization manifold

The tangent spaces play a critical role in the error of Neural Galerkin schemes (Zhang et al., 2024). To see this, let us assume there exists an $\epsilon : [0, T] \to [0, \infty)$ that bounds the projection error of the right-hand side function f onto the tangent space $\mathcal{T}_{\tilde{q}(\boldsymbol{\theta}(t),\cdot)}\mathcal{M}_\Theta$ at the current field $\tilde{q}(\boldsymbol{\theta}(t), \cdot)$,

$$\|f(t, \cdot, \tilde{q}(\boldsymbol{\theta}(t), \cdot)) - \mathrm{P}_{\boldsymbol{\theta}(t)} f(t, \cdot, \tilde{q}(\boldsymbol{\theta}(t), \cdot))\|_\nu \leq \epsilon(t)\,.$$

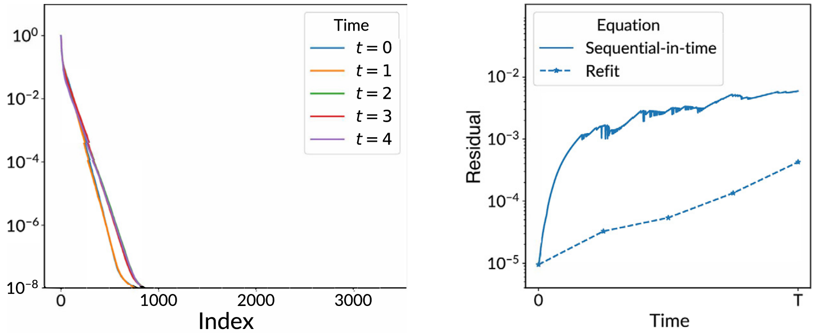
Under standard assumptions (Lubich, 2005, 2008; Lasser and Lubich, 2020; Zhang et al., 2024) on f and $\tilde{q}$, the error in the Neural Galerkin solution is bounded as

$$\|q(t, \cdot) - \tilde{q}(\boldsymbol{\theta}(t), \cdot)\|_\nu \leq \mathrm{e}^{Ct}\left(e_0 + \int_0^t \mathrm{e}^{-Cs}\epsilon(s)\mathrm{d}s\right), \tag{28}$$

where $C > 0$ is a constant independent of the time t and e_0 is the error in representing the initial condition in the parametrization,

$$e_0 = \|q(0, \cdot) - \tilde{q}(\boldsymbol{\theta}(0), \cdot)\|_\nu\,.$$

The bound (28) shows that the error is driven by the bound on the projection error $\epsilon(t)$ of projecting the right-hand side function onto the tangent space. The bound (28) underlines the importance of the tangent spaces.



(a) low-rank batch gradients (b) local approximations versus re-fitting from scratch

FIGURE 7 The low-rankness of the batch gradient $\boldsymbol{J}_t$ in Neural Galerkin schemes motivates a randomized sparse version of Neural Galerkin (Berman and Peherstorfer, 2023) that updates only a subset of $s \ll p$ of the total number of p parameters in the deep network at each time step, which leads to speedups and has a regularization effect. (Figure from Berman and Peherstorfer, 2023.)

5.2 Tangent space collapse

Nonlinear parametrizations tend to lead to low-rank batch gradients $\boldsymbol{J}_t \in \mathbb{R}^{m \times p}$ as defined in (26). The space spanned by the columns of $\boldsymbol{J}_t$ therefore is of lower rank than $\min\{m, p\}$, which is a phenomenon that is referred to as tangent space collapse in Zhang et al. (2024); see Fig. 7a. Consequently, if the right-hand side function (or the batch right-hand side function (27)) cannot be represented well anymore in the spanned space, then this leads to a quick deterioration of the accuracy of Neural Galerkin solutions because of the bound given in (28).

The low-rankness of the batch gradient additionally means that there are multiple trajectories $\boldsymbol{\theta}(t)$ that solve the dynamical system (16) because the matrix $M(t, \boldsymbol{\theta}(t))$ can become singular. Adding a regularization term can enforce a unique solution; however, it is unclear if such a regularization can prevent the collapsing tangent space phenomenon and thus avoid the loss of expressiveness. In particular, only local moves in the parameter domain are made via the time stepping rather than global jumps, which means the scheme can get stuck in poor parameter regions despite regularization. Detailed discussions about the collapsing tangent phenomena and local moves are provided in Zhang et al. (2024); Berman and Peherstorfer (2023).

5.3 Leveraging low-rankness of batch gradients with subsampling

Let us recall the least-squares problem from (23), the batch gradient $\boldsymbol{J}_t(\boldsymbol{\theta}(t))$ defined in (26), and the batch right-hand side $\boldsymbol{f}_t(\boldsymbol{\theta}(t))$ given in (27). In continuous time, we obtain the least-squares problem

$$\min_{\dot{\boldsymbol{\theta}}(t) \in \mathbb{R}^p} \|\boldsymbol{J}_t(\boldsymbol{\theta}(t))\dot{\boldsymbol{\theta}}(t) - \boldsymbol{f}_t(\boldsymbol{\theta}(t))\|_2 \,, \tag{29}$$

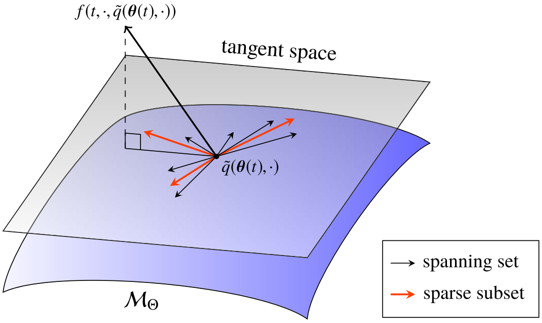

$f(t, \cdot, \tilde{q}(\boldsymbol{\theta}(t), \cdot))$
tangent space
$\tilde{q}(\boldsymbol{\theta}(t), \cdot)$
$\mathcal{M}_\Theta$
→ spanning set
→ sparse subset

FIGURE 8 Randomized sparse Neural Galerkin schemes update only a subset of $s \ll p$ parameters of the deep network at each time step, which is motivated by the low-rankness of the batch gradient. (Figure from Berman and Peherstorfer, 2023.)

where the batch gradient and the batch right-hand side are obtained by sampling from the measure ν. The low-rankness of the batch gradient $\boldsymbol{J}_t$ means that considering a subset of the columns of $\boldsymbol{J}_t$ is sufficient in the least-squares problem (29); see Fig. 8. Motivated by this, the work by Berman and Peherstorfer (2023) proposes to update only $s \ll p$ components of $\boldsymbol{\theta}(t)$ over time t.

Consider therefore the subsampled parameter vector

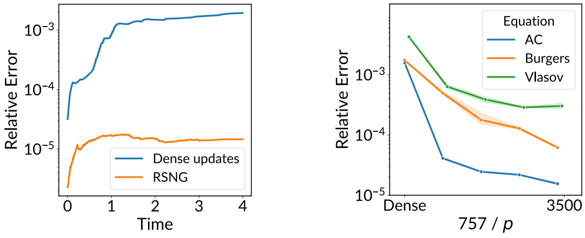
$$\boldsymbol{\theta}_s(t) = \boldsymbol{S}_t^T \boldsymbol{\theta}(t),$$

where $\boldsymbol{S}_t \in \mathbb{R}^{p \times s}$ subselects s components out of the p components of $\boldsymbol{\theta}(t)$. The subsampled parameter vector is called $\boldsymbol{\theta}_s(t) \in \mathbb{R}^s$. It is then proposed in Berman and Peherstorfer (2023) to solve the sketched least-squares problem

$$\min_{\dot{\boldsymbol{\theta}}_s(t) \in \mathbb{R}^s} \|\boldsymbol{J}_t(\boldsymbol{\theta}(t)) \boldsymbol{S}_t \dot{\boldsymbol{\theta}}_s(t) - \boldsymbol{f}_t(t, \cdot, \tilde{q}(\boldsymbol{\theta}(t), \cdot))\|_\nu, \tag{30}$$

where the sketching happens over the parameter vector $\boldsymbol{\theta}(t)$. Notice that the columns of the batch gradient that correspond to indices of the parameter vector $\boldsymbol{\theta}(t)$ that are not selected can be ignored because they are multiplied with zeros in the objective of (30). It is found in Berman and Peherstorfer (2023) that one way to prevent the tangent spaces from collapsing is to randomly select the s components that are updated over time t. Thus, the matrix $\boldsymbol{S}_t$ becomes a random matrix that uniformly draws s out of p components of $\boldsymbol{\theta}(t)$.

Updating only $s \ll p$ components per time step leads to lower runtimes because the number of unknowns in the randomly sketched least-squares problem (30) is lower than in the least-squares problem (29) that densely updates all components of $\boldsymbol{\theta}(t)$. In particular, if a direct, dense numerical linear algebra method is used to solve the least-squares problems such as based on the singular value decomposition or the QR decomposition, then the speedup scales quadratically in $1/s$.



(a) comparison to dense updates (b) larger ambient networks

FIGURE 9 Randomized sparse Neural Galerkin schemes achieve orders of magnitude lower errors because the randomization of the parameters that are updated has a regularization effect, as plot (a) shows. Even though the number of parameters $s \ll p$ that is updated at each time step is low, randomized sparse Neural Galerkin schemes leverage the increasing expressiveness as the total number of parameters p of the network is increased, as plot (b) shows. (Figure from Berman and Peherstorfer, 2023.)

5.4 Numerical experiments with randomized sparse Neural Galerkin schemes

We report two numerical experiments from Berman and Peherstorfer (2023). Let us first consider the Allen-Cahn equation with a quadratic potential over a one-dimensional spatial domain as well as a fully connected feedforward network with rational activation functions; the details of the setup are given in Berman and Peherstorfer (2023). Fig. 9a compares the error obtained with a three-layer network for which all parameters are updated at each time step (“dense”) versus a seven-layer network where only a sparse subset of the parameters are updated so that the size of the sparse subset is the same as the total number of parameters in the three-layer network. The plot in Fig. 9a shows that two orders of magnitude improvement in the relative error is achieved with the sparse updates compared to dense updates. Fig. 9b keeps the number of parameters s that are updated per time step fixed at $s = 757$ but increases the number of layers in the network from which the s parameters are taken. The plot shows that the error decays and thus the increasing expressiveness of the larger, ambient network can be leveraged even though the same number of parameters s are updated at each time step.

Besides reducing the error with sparse updates, we also obtain speedups compared to updating all parameters in the network; see Fig. 10. The bars with label “direct” correspond to dense updates with direct least-squares solvers based on the singular value decomposition. The bars with “iterative” correspond to iterative methods as described in Berman and Peherstorfer (2023). And the bars with “RSNG” show sparse updates with a direct least-squares solver, which achieves the lowest runtime by orders of magnitude.

6 Conclusions

Nonlinear parametrizations have been shown to achieve faster best-approximation error decays than linear parametrizations, which can be interpreted as them

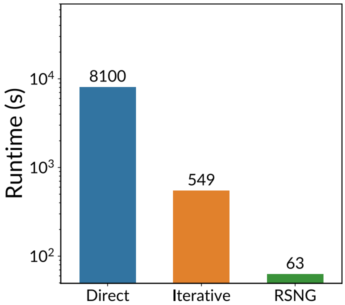


FIGURE 10 Because randomized sparse Neural Galerkin schemes update only $s \ll p$ out of the total of p parameters of the deep network, speedups of almost two orders of magnitude can be achieved compared to dense updates with direct least-squares solvers and about one order of magnitude compared to iterative least-squares solvers. (Figure from Berman and Peherstorfer, 2023.)

being more expressiveness with the same number of parameters than linear parametrizations. However, from a computational mathematics and numerical analysis perspective, major challenges remain regarding the development and analysis of numerical methods that can leverage that increased expressiveness and realize it numerically. In this survey, we discussed Neural Galerkin schemes that are motivated by the success of the method of lines from numerical analysis with linear parametrizations. The key building block is the Dirac-Frenkel variational principle to derive dynamical systems for the time-dependent, finite-dimensional parameter vectors of the nonlinear parametrizations. Another key ingredient is adaptive sampling to efficiently evaluate the residual objective. The numerical results in this survey and in the original publications where they have been presented first indicate that sequential-in-time approaches can be beneficial in terms of inherently providing causal numerical solutions, enforcing physics constraints, and requiring fewer parameters to well approximate solution fields compared to global-in-time methods. However, major challenges lie ahead to develop numerical methods for nonlinear parametrizations that are as rigorously analyzable, stable, robust, and efficient as today's numerical methods for linear parametrizations.

References

Amsallem, D., Zahr, M.J., Farhat, C., 2012. Nonlinear model order reduction based on local reduced-order bases. International Journal for Numerical Methods in Engineering 92 (10), 891–916.

Anderson, W., Farazmand, M., 2022. Evolution of nonlinear reduced-order solutions for PDEs with conserved quantities. SIAM Journal on Scientific Computing 44 (1), A176–A197.

Antoulas, A.C., 2005. Approximation of Large-Scale Dynamical Systems. SIAM.

Antoulas, A.C., Beattie, C.A., Gugercin, S., 2021. Interpolatory Methods for Model Reduction. SIAM.

Arbes, F., Greif, C., Urban, K., 2023. The Kolmogorov N-width for linear transport: exact representation and the influence of the data. arXiv:2305.00066.

Bar-Sinai, Y., Hoyer, S., Hickey, J., Brenner, M.P., 2019. Learning data-driven discretizations for partial differential equations. Proceedings of the National Academy of Sciences 116 (31), 15344–15349.

Barnett, J., Farhat, C., 2022. Quadratic approximation manifold for mitigating the Kolmogorov barrier in nonlinear projection-based model order reduction. Journal of Computational Physics 464, 111348.

Benner, P., Gugercin, S., Willcox, K., 2015. A survey of projection-based model reduction methods for parametric dynamical systems. SIAM Review 57 (4), 483–531.

Berg, J., Nyström, K., 2018. A unified deep artificial neural network approach to partial differential equations in complex geometries. Neurocomputing 317, 28–41.

Berger, M., Colella, P., 1989. Local adaptive mesh refinement for shock hydrodynamics. Journal of Computational Physics 82 (1), 64–84.

Berger, M.J., LeVeque, R.J., 1998. Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems. SIAM Journal on Numerical Analysis 35, 2298–2316.

Berman, J., Peherstorfer, B., 2023. Randomized sparse Neural Galerkin schemes for solving evolution equations with deep networks. In: Thirty-Seventh Conference on Neural Information Processing Systems.

Berman, J., Peherstorfer, B., 2024. CoLoRA: continuous low-rank adaptation for reduced implicit neural modeling of parameterized partial differential equations. arXiv:2402.14646.

Billaud-Friess, M., Nouy, A., 2017. Dynamical model reduction method for solving parameter-dependent dynamical systems. SIAM Journal on Scientific Computing 39 (4), A1766–A1792.

Binev, P., Cohen, A., Dahmen, W., DeVore, R., Petrova, G., Wojtaszczyk, P., 2011. Convergence rates for greedy algorithms in reduced basis methods. SIAM Journal on Mathematical Analysis 43 (3), 1457–1472.

Black, Felix, Schulze, Philipp, Unger, Benjamin, 2020. Projection-based model reduction with dynamically transformed modes. ESAIM: M2AN 54 (6), 2011–2043.

Bruna, J., Peherstorfer, B., Vanden-Eijnden, E., 2024. Neural Galerkin schemes with active learning for high-dimensional evolution equations. Journal of Computational Physics 496, 112588.

Buffa, A., Maday, Y., Patera, A.T., Prud'homme, C., Turinici, G., 2012. A priori convergence of the greedy algorithm for the parametrized reduced basis method. ESAIM: M2AN 46 (3), 595–603.

Cagniart, N., Maday, Y., Stamm, B., 2019. Model order reduction for problems with large convection effects. In: Chetverushkin, B.N., Fitzgibbon, W., Kuznetsov, Y., Neittaanmäki, P., Periaux, J., Pironneau, O. (Eds.), Contributions to Partial Differential Equations and Applications. Springer International Publishing, Cham, pp. 131–150.

Carlberg, K., 2015. Adaptive h-refinement for reduced-order models. International Journal for Numerical Methods in Engineering 102 (5), 1192–1210.

Cohen, A., DeVore, R., 2016. Kolmogorov widths under holomorphic mappings. IMA Journal of Numerical Analysis 36 (1), 1–12.

Cohen, A., DeVore, R., Petrova, G., Wojtaszczyk, P., 2022. Optimal stable nonlinear approximation. Foundations of Computational Mathematics 22 (3), 607–648.

Cohen, Albert, Dahmen, Wolfgang, DeVore, Ronald, Nichols, James, 2020. Reduced basis greedy selection using random training sets. ESAIM: M2AN 54 (5), 1509–1524.

Daubechies, I., DeVore, R., Foucart, S., Hanin, B., Petrova, G., 2022. Nonlinear approximation and (deep) ReLU networks. Constructive Approximation 55 (1), 127–172.

DeVore, R., Hanin, B., Petrova, G., 2021. Neural network approximation. Acta Numerica 30, 327–444.

DeVore, R.A., Howard, R., Micchelli, C., 1989. Optimal nonlinear approximation. Manuscripta Mathematica 63 (4), 469–478.

DeVore, R.A., Kyriazis, G., Leviatan, D., Tikhomirov, V.M., 1993. Wavelet compression and nonlinear n-widths. Advances in Computational Mathematics 1 (2), 197–214.

Dihlmann, M., Drohmann, M., Haasdonk, B., 2011. Model reduction of parametrized evolution problems using the reduced basis method with adaptive time-partitioning. In: Proc. of ADMOS 2011.

Dirac, P.A.M., 1930. Note on exchange phenomena in the Thomas atom. Mathematical Proceedings of the Cambridge Philosophical Society 26 (3), 376–385.

Dissanayake, M.W.M.G., Phan-Thien, N., 1994. Neural-network-based approximations for solving partial differential equations. Communications in Numerical Methods in Engineering 10 (3), 195–201.

Du, Y., Zaki, T.A., 2021. Evolutional deep neural network. Physical Review E 104 (4).

E, W., Han, J., Jentzen, A., 2017. Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. Communications in Mathematics and Statistics 5 (4), 349–380.

Eftang, J.L., Stamm, B., 2012. Parameter multi-domain 'hp' empirical interpolation. International Journal for Numerical Methods in Engineering 90 (4), 412–428.

Ehrlacher, V., Lombardi, D., Mula, O., Vialard, F.-X., 2020. Nonlinear model reduction on metric spaces. Application to one-dimensional conservative PDEs in Wasserstein spaces. ESAIM: M2AN 54 (6), 2159–2197.

Einkemmer, L., Hu, J., Wang, Y., 2021. An asymptotic-preserving dynamical low-rank method for the multi-scale multi-dimensional linear transport equation. Journal of Computational Physics 439, 110353.

Einkemmer, L., Lubich, C., 2019. A quasi-conservative dynamical low-rank algorithm for the Vlasov equation. SIAM Journal on Scientific Computing 41 (5), B1061–B1081.

Ern, A., Guermond, J.-L., 2004. Theory and Practice of Finite Elements. Springer.

Finzi, M.A., Potapczynski, A., Choptuik, M., Wilson, A.G., 2023. A stable and scalable method for solving initial value PDEs with neural networks. In: The Eleventh International Conference on Learning Representations.

Frenkel, J., 1934. Wave Mechanics, Advanced General Theory. Clarendon Press, Oxford.

Geelen, R., Balzano, L., Wright, S., Willcox, K., 2024. Learning physics-based reduced-order models from data using nonlinear manifolds. Chaos 34 (3), 033122.

Geelen, R., Willcox, K., 2022. Localized non-intrusive reduced-order modelling in the operator inference framework. Philosophical Transactions - Royal Society. Mathematical, Physical and Engineering Sciences 380 (2229), 20210206.

Geelen, R., Wright, S., Willcox, K., 2023. Operator inference for non-intrusive model reduction with quadratic manifolds. Computer Methods in Applied Mechanics and Engineering 403, 115717.

Gerbeau, J.-F., Lombardi, D., 2014. Approximated Lax pairs for the reduced order integration of nonlinear evolution equations. Journal of Computational Physics 265, 246–269.

Greif, C., Urban, K., 2019. Decay of the Kolmogorov N-width for wave problems. Applied Mathematics Letters 96, 216–222.

Han, J., Jentzen, A., E, W., 2018. Solving high-dimensional partial differential equations using deep learning. Proceedings of the National Academy of Sciences 115 (34), 8505–8510.

Hesthaven, J.S., Pagliantini, C., Rozza, G., 2022. Reduced basis methods for time-dependent problems. Acta Numerica 31, 265–345.

Huang, C., Duraisamy, K., 2023. Predictive reduced order modeling of chaotic multi-scale problems using adaptively sampled projections. Journal of Computational Physics 491, 112356.

Iollo, A., Lombardi, D., 2014. Advection modes by optimal mass transfer. Physical Review E 89, 022923.

Issan, O., Kramer, B., 2023. Predicting solar wind streams from the inner-heliosphere to earth via shifted operator inference. Journal of Computational Physics 473, 111689.

Jens, D.J.K., Eftang, L., Patera, A.T., 2011. An hp certified reduced basis method for parametrized parabolic partial differential equations. Mathematical and Computer Modelling of Dynamical Systems 17 (4), 395–422.

Kast, M., Hesthaven, J.S., 2023. Positional embeddings for solving PDEs with evolutional deep neural networks. arXiv:2308.03461.

Kaulmann, S., Flemisch, B., Haasdonk, B., Lie, K.A., Ohlberger, M., 2015. The localized reduced basis multiscale method for two-phase flows in porous media. International Journal for Numerical Methods in Engineering 102 (5), 1018–1040.

Khoo, Y., Lu, J., Ying, L., 2018. Solving for high-dimensional committor functions using artificial neural networks. Research in the Mathematical Sciences 6 (1), 1.

Kim, Y., Choi, Y., Widemann, D., Zohdi, T., 2022. A fast and accurate physics-informed neural network reduced order model with shallow masked autoencoder. Journal of Computational Physics 451, 110841.

Koch, O., Lubich, C., 2007. Dynamical low-rank approximation. SIAM Journal on Matrix Analysis and Applications 29 (2), 434–454.

Kochkov, D., Smith, J.A., Alieva, A., Wang, Q., Brenner, M.P., Hoyer, S., 2021. Machine learning–accelerated computational fluid dynamics. Proceedings of the National Academy of Sciences 118 (21).

Kramer, B., Peherstorfer, B., Willcox, K.E., 2024. Learning nonlinear reduced models from data with operator inference. Annual Review of Fluid Mechanics 56 (1), 521–548.

Lasser, C., Lubich, C., 2020. Computing quantum dynamics in the semiclassical regime. Acta Numerica 29, 229–401.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553), 436–444.

Lee, K., Carlberg, K.T., 2020. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. Journal of Computational Physics 404, 108973.

Li, Q., Lin, B., Ren, W., 2019. Computing committor functions for the study of rare events using deep learning. Journal of Chemical Physics 151 (5), 054112.

Liu, Q., Wang, D., 2016. Stein variational gradient descent: a general purpose Bayesian inference algorithm. In: Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R. (Eds.), Advances in Neural Information Processing Systems, vol. 29. Curran Associates, Inc., pp. 2378–2386.

Lubich, C., 2005. On variational approximations in quantum molecular dynamics. Mathematics of Computation 74 (250), 765–779.

Lubich, C., 2008. From Quantum to Classical Molecular Dynamics: Reduced Models and Numerical Analysis, vol. 12. European Mathematical Society.

Maday, Y., Patera, A.T., Turinici, G., 2002. Global a priori convergence theory for reduced-basis approximations of single-parameter symmetric coercive elliptic partial differential equations. Comptes Rendus. Mathématique 335 (3), 289–294.

Maday, Y., Stamm, B., 2013. Locally adaptive greedy approximations for anisotropic parameter reduced basis spaces. SIAM Journal on Scientific Computing 35 (6), A2417–A2441.

Musharbash, E., Nobile, F., 2017. Symplectic dynamical low rank approximation of wave equations with random parameters. Mathicse Technical Report nr 18.2017.

Musharbash, E., Nobile, F., 2018. Dual dynamically orthogonal approximation of incompressible Navier Stokes equations with random boundary conditions. Journal of Computational Physics 354, 135–162.

Musharbash, E., Nobile, F., Zhou, T., 2015. Error analysis of the dynamically orthogonal approximation of time dependent random pdes. SIAM Journal on Scientific Computing 37 (2), A776–A810.

Ohlberger, M., Rave, S., 2013. Nonlinear reduced basis approximation of parameterized evolution equations via the method of freezing. Comptes Rendus. Mathématique 351 (23), 901–906.

Ohlberger, M., Rave, S., 2016. Reduced basis methods: success, limitations and future challenges. In: Proceedings of the Conference Algoritmy, pp. 1–12.

Orús, R., 2019. Tensor networks for complex quantum systems. Nature Reviews Physics 1 (9), 538–550.

Papapicco, D., Demo, N., Girfoglio, M., Stabile, G., Rozza, G., 2022. The neural network shifted-proper orthogonal decomposition: a machine learning approach for non-linear reduction of hyperbolic equations. Computer Methods in Applied Mechanics and Engineering 392, 114687.

Peherstorfer, B., 2020. Model reduction for transport-dominated problems via online adaptive bases and adaptive sampling. SIAM Journal on Scientific Computing 42, A2803–A2836.

Peherstorfer, B., 2022. Breaking the Kolmogorov barrier with nonlinear model reduction. Notices of the American Mathematical Society 69, 725–733.

Peherstorfer, B., Butnaru, D., Willcox, K., Bungartz, H.-J., 2014. Localized discrete empirical interpolation method. SIAM Journal on Scientific Computing 36 (1), A168–A192.

Peherstorfer, B., Willcox, K., 2015. Online adaptive model reduction for nonlinear systems via low-rank updates. SIAM Journal on Scientific Computing 37 (4), A2123–A2150.

Pinkus, A., 1985. n-Widths in Approximation Theory. Springer, Berlin, Heidelberg.

Qian, E., Kramer, B., Peherstorfer, B., Willcox, K., 2020. Lift & learn: physics-informed machine learning for large-scale nonlinear dynamical systems. Physica D. Nonlinear Phenomena 406, 132401.

Raissi, M., Perdikaris, P., Karniadakis, G., 2019. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. Journal of Computational Physics 378, 686–707.

Ramezanian, D., Nouri, A.G., Babaee, H., 2021. On-the-fly reduced order modeling of passive and reactive species via time-dependent manifolds. Computer Methods in Applied Mechanics and Engineering 382, 113882.

Reiss, J., Schulze, P., Sesterhenn, J., Mehrmann, V., 2018. The shifted proper orthogonal decomposition: a mode decomposition for multiple transport phenomena. SIAM Journal on Scientific Computing 40 (3), A1322–A1344.

Romor, F., Stabile, G., Rozza, G., 2023. Non-linear manifold reduced-order models with convolutional autoencoders and reduced over-collocation method. Journal of Scientific Computing 94 (3), 74.

Rotskoff, G.M., Mitchell, A.R., Vanden-Eijnden, E., 2022. Active importance sampling for variational objectives dominated by rare events: consequences for optimization and generalization. In: Bruna, J., Hesthaven, J., Zdeborova, L. (Eds.), Proceedings of the 2nd Mathematical and Scientific Machine Learning Conference. In: Proceedings of Machine Learning Research, vol. 145. PMLR, pp. 757–780.

Rowley, C.W., Colonius, T., Murray, R.M., 2004. Model reduction for compressible flows using POD and Galerkin projection. Physica D. Nonlinear Phenomena 189 (1), 115–129.

Rowley, C.W., Marsden, J.E., 2000. Reconstruction equations and the Karhunen–Loève expansion for systems with symmetry. Physica D. Nonlinear Phenomena 142 (1), 1–19.

Rozza, G., Huynh, D.B.P., Patera, A.T., 2008. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. Archives of Computational Methods in Engineering 15 (3), 229–275.

Rudy, S.H., Nathan Kutz, J., Brunton, S.L., 2019. Deep learning of dynamics and signal-noise decomposition with time-stepping constraints. Journal of Computational Physics 396, 483–506.

Sapsis, T.P., Lermusiaux, P.F., 2009. Dynamically orthogonal field equations for continuous stochastic dynamical systems. Physica D. Nonlinear Phenomena 238 (23), 2347–2360.

Schwerdtner, P., Peherstorfer, B., 2024. Greedy construction of quadratic manifolds for nonlinear dimensionality reduction and nonlinear model reduction. arXiv:2403.06732.

Schwerdtner, P., Schulze, P., Berman, J., Peherstorfer, B., 2023. Nonlinear embeddings for conserving Hamiltonians and other quantities with Neural Galerkin schemes. arXiv:2310.07485.

Sharma, H., Mu, H., Buchfink, P., Geelen, R., Glas, S., Kramer, B., 2023. Symplectic model reduction of Hamiltonian systems using data-driven quadratic manifolds. Computer Methods in Applied Mechanics and Engineering 417, 116402.

Singh, R., Uy, W., Peherstorfer, B., 2023. Lookahead data-gathering strategies for online adaptive model reduction of transport-dominated problems. Chaos.

Sirignano, J., Spiliopoulos, K., 2018. DGM: a deep learning algorithm for solving partial differential equations. Journal of Computational Physics 375, 1339–1364.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15 (56), 1929–1958.

Sung, Y.-L., Nair, V., Raffel, C., 2021. Training neural networks with fixed sparse masks. arXiv: 2111.09839.

Taddei, T., 2020. A registration method for model order reduction: data compression and geometry reduction. SIAM Journal on Scientific Computing 42 (2), A997–A1027.

Taddei, T., Perotto, S., Quarteroni, A., 2015. Reduced basis techniques for nonlinear conservation laws. ESAIM: M2AN 49 (3), 787–814.

Taddei, Tommaso, Zhang, Lei, 2021. Space-time registration-based model reduction of parameterized one-dimensional hyperbolic pdes. ESAIM: M2AN 55 (1), 99–130.

Uy, W., Wentland, C., Huang, C., Peherstorfer, B., 2024. Reduced models with nonlinear approximations of latent dynamics for model premixed flame problems. In: Rozza, G., Stabile, G., Gunzburger, M., D'Elia, M. (Eds.), Reduction, Approximation, Machine Learning, Surrogates, Emulators and Simulators. In: Lecture Notes in Computational Science and Engineering, vol. 151. Springer. In press.

Vapnik, V., 1991. Principles of risk minimization for learning theory. In: Moody, J., Hanson, S., Lippmann, R. (Eds.), Advances in Neural Information Processing Systems, vol. 4. Morgan-Kaufmann.

Verwer, J.G., Sanz-Serna, J.M., 1984. Convergence of method of lines approximations to partial differential equations. Computing 33 (3), 297–313.

Wang, Q., Ripamonti, N., Hesthaven, J.S., 2020. Recurrent neural network closure of parametric POD-Galerkin reduced-order models based on the Mori-Zwanzig formalism. Journal of Computational Physics 410, 109402.

Wen, Y., Vanden-Eijnden, E., Peherstorfer, B., 2024. Coupling parameter and particle dynamics for adaptive sampling in Neural Galerkin schemes. Physica D.

Zafarullah, A., 1970. Application of the method of lines to parabolic partial differential equations with error estimates. Journal of the ACM 17 (2), 294–302.

Zahr, M.J., Farhat, C., 2015. Progressive construction of a parametric reduced-order model for PDE-constrained optimization. International Journal for Numerical Methods in Engineering 102 (5), 1111–1135.

Zaken, E.B., Ravfogel, S., Goldberg, Y., 2022. BitFit: simple parameter-efficient fine-tuning for transformer-based masked language-models. arXiv:2106.10199.

Zhang, H., Chen, Y., Vanden-Eijnden, E., Peherstorfer, B., 2024. Sequential-in-time training of nonlinear parametrizations for solving time-dependent partial differential equations. arXiv:2404.01145.

Zimmermann, R., Peherstorfer, B., Willcox, K., 2018. Geometric subspace updates with applications to online adaptive nonlinear model reduction. SIAM Journal on Matrix Analysis and Applications 39 (1), 234–261.