CoLoRA: Continuous Low-Rank Adaptation for Reduced Implicit Neural Modeling of Parameterized Partial Differential Equations

Jules Berman [1] **Benjamin Peherstorfer** [1]

Abstract

This work introduces reduced models based on Continuous Low Rank Adaptation (CoLoRA) that pre-train neural networks for a given partial differential equation and then continuously adapt low-rank weights in time to rapidly predict the evolution of solution fields at new physics parameters and new initial conditions. The adaptation can be either purely data-driven or via an equation-driven variational approach that provides Galerkin-optimal approximations. Because CoLoRA approximates solution fields locally in time, the rank of the weights can be kept small, which means that only few training trajectories are required offline so that CoLoRA is well suited for data-scarce regimes. Predictions with CoLoRA are orders of magnitude faster than with classical methods and their accuracy and parameter efficiency is higher compared to other neural network approaches.

1. Introduction

Many phenomena of interest in science and engineering depend on physics parameters $\boldsymbol{\mu}$ that influence the temporal and spatial evolution of the system such as the Reynolds number in fluid mechanics and conductivity coefficients in heat transfer. Rapidly simulating physical phenomena for a large sample $M \gg 1$ of physics parameters $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_M$ is paramount in science and engineering, e.g., for finding optimal designs, inverse problems, data assimilation, uncertainty quantification, and control. Numerically solving the underlying parameterized partial differential equations (PDEs) with standard numerical methods (Hughes, 2012; LeVeque, 2002) for large numbers of different physics parameters is prohibitively expensive in many applications and thus one often resorts to reduced modeling.

[1]Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, USA. Correspondence to: Jules Berman <jmb1174@nyu.edu>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

The Kolmogorov barrier Reduced models exploit structure in PDE problems to more efficiently approximate solution fields. The conventional structure that is leveraged is low rankness in the sense of the classical principal component analysis. Reduced models based on such low-rank approximations can achieve exponentially fast error decays e^{-cn} with the rank n for a wide range of (mostly nicely behaved elliptic) problems (Maday et al., 2002; Cohen & DeVore, 2016). However, linear low-rank approximations are affected by the so-called Kolmogorov barrier, which states that there are classes of PDEs for which linear approximations have an error decay rate of at best $1/\sqrt{n}$ (Ohlberger & Rave, 2016; Greif & Urban, 2019). Examples of PDE classes that are affected by the Kolmogorov barrier are often describing transport-dominated phenomena such as strongly advecting flows and wave-like behavior; see (Peherstorfer, 2022) for a survey.

Our contribution: nonlinear reduced modeling with continuous low-rank adaptation (CoLoRA) In this work, we build on LoRA (Low Rank Adaptation) for developing parameterizations for reduced models (Hu et al., 2022). LoRA has been developed for fine-tuning large language models and leverages the observation that fine-tuning objectives can be efficiently optimized on low-dimensional parameter spaces (Li et al., 2018; Aghajanyan et al., 2021; Hu et al., 2022). This has been observed not only for large language models but also when approximating solution fields of physical phenomena (Bachmayr et al., 2017; Grasedyck et al., 2013; Berman & Peherstorfer, 2023). We build on the pre-training/fine-tuning paradigm of LoRA but modify it to Continuous LoRA (CoLoRA) that reflects our PDE setting by allowing continuous-in-time adaptation ("fine-tuning") of parts of the low-rank components as the solution fields of the PDEs evolve; see Figure 1. This inductive bias is in agreement with how typically physical phenomena evolve over time, namely smoothly and along a latent low-dimensional structure. By composing multiple CoLoRA layers in a deep network, we obtain a nonlinear parameterization that can circumvent the Kolmogorov barrier while having a pre-training/fine-tuning decomposition with an inductive bias that reflects the special meaning of the time variable t in PDE problems; see Figure 2.

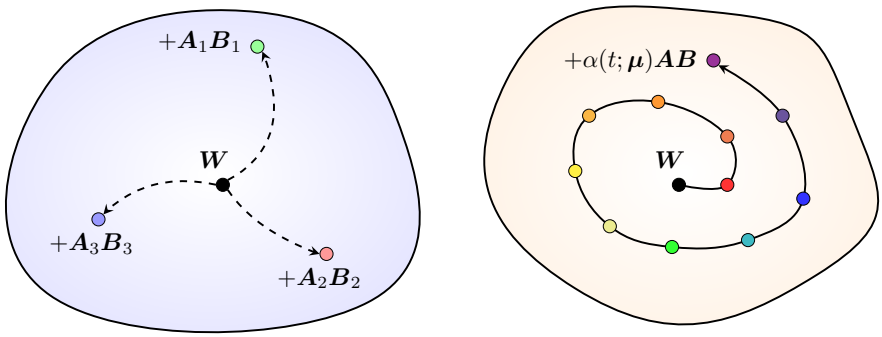

Figure 1. LoRA fine-tunes networks to downstream tasks by adapting low-rank matrices $\boldsymbol{AB}$. Our CoLoRA introduces a scaling $\alpha(t, \boldsymbol{\mu})$ on the low-rank matrix $\boldsymbol{AB}$ to adapt networks continuously to predict PDE solution trajectories.

The time variable in CoLoRA models Time enters CoLoRA models via the low-rank network weights that are adapted in the online ("fine-tuning") phase of reduced modeling, rather than as input to the network as the spatial coordinates. Treating time separately from inputs such as the spatial coordinates aligns well with approximating PDE solution fields sequentially in time, rather than globally over the whole time-space domain. The sequential-in-time approximation paradigm of CoLoRA allows keeping the number of offline and online parameters low, because the solution field is approximated only locally in time. Requiring a low number of online and offline parameters indicates that the inductive bias induced by separating out the time variable from the spatial coordinate is in agreement with many physics problems. Furthermore, the CoLoRA architecture can be trained on low numbers of training trajectories compared to, e.g., operator learning (Li et al., 2021; Lu et al., 2021), which is important because standard numerical simulations can be expensive and thus only a limited number of training trajectories are available in many cases. Having network weights that depend on time allows combining CoLoRA with equation-driven variational approaches (Lasser & Lubich, 2020; Du & Zaki, 2021; Anderson & Farazmand, 2022; Bruna et al., 2024; Wen et al., 2024) to obtain reduced solutions that are Galerkin-optimal, which opens the door to analyses, error bounds, and goes far beyond purely data-driven forecasting.

Literature review There are purely data-driven surrogate modeling methods such as operator learning (Li et al., 2021; Lu et al., 2021; Boullé & Townsend, 2023) that can require large amounts of data because they aim to learn a generic operator map over the full model space. Model reduction (Antoulas, 2005; Rozza et al., 2008; Benner et al., 2015; Antoulas et al., 2021; Kramer et al., 2024) considers a more structured problem via the physics parameter $\boldsymbol{\mu}$, for which nonlinear model reduction methods based on autoencoders have been presented in (Lee & Carlberg, 2020; 2021; Kim et al., 2022; Romor et al., 2023); however, they can require going back to the high-fidelity numerical model to drive the dynamics, which can be expensive. Alternative approaches learn the low-dimensional latent dynamics (Fulton et al., 2019; Lee & Parish, 2021; Wan et al., 2023). Additional literature of nonlinear model reduction is reviewed in Appendix A.

There is a range of methods based on implicit neural representations that updates the network parameters either based on the equation (Chen et al., 2023a;b) or via hypernetworks (Pan et al., 2023; Yin et al., 2023); these methods are closest to CoLoRA, except that we will show that CoLoRA's low-rank adaptation achieves lower errors with lower parameter counts in our examples. There are pretraining/fine-tuning techniques for global-in-time methods such as physics-informed neural networks (Raissi et al., 2019) that are purely data-driven once trained. Some of these approaches use hyper-networks (de Avila Belbute-Peres et al., 2021; Cho et al., 2023) and other meta-learning such as Finn et al. (2017). But typically these approaches are over time-space domains and thus do not make a special treatment of time or the PDE dynamics, which is a key feature of CoLoRA. Adaptive low-rank approximations have been used in scientific computing for a long time (Koch & Lubich, 2007; Sapsis & Lermusiaux, 2009; Peherstorfer & Willcox, 2015; Einkemmer & Lubich, 2019); however, they use one layer only and adapt the low-rank matrices directly with time. We have a more restricted adaptation in time that has fewer parameters, which is sufficient due to the nonlinear composition of multiple layers. Other low-rank approximations have been widely used in the context of deep networks (Sainath et al., 2013; Zhang et al., 2014; Zhao et al., 2016; Khodak et al., 2021; Schotthöfer et al., 2022) but not in PDE settings.

Summary CoLoRA leverages that PDE dynamics are typically continuous in time while evolving on low-dimensional manifolds. With CoLoRA, we achieve nonlinear param-

eterizations that circumvent the Kolmogovorv barrier of transport-dominated problems and can provide predictions purely data-driven or in a variational sense using the PDEs. Our numerical experiments show that CoLoRA requires a low number of training trajectories, achieves orders of magnitude speedups compared to classical methods, and outperforms the existing state-of-the-art neural-network-based model reduction methods in parameter count and accuracy on a wide variety of PDE problems.

2. Parameterized PDEs

Let $u : \mathcal{T} \times \Omega \times \mathcal{D} \to \mathbb{R}$ be a solution field that represents, e.g., temperature, density, velocities, or pressure of a physical process. The solution field u depends on time $t \in \mathcal{T} = [0, T) \subset \mathbb{R}$, spatial coordinate $\boldsymbol{x} \in \Omega \subset \mathbb{R}^d$, and physics parameter $\boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^{d'}$. The solution field u is governed by a parameterized PDE,

$$\begin{aligned} \partial_t u(t, \boldsymbol{x}; \boldsymbol{\mu}) &= f(\boldsymbol{x}, u; \boldsymbol{\mu}) \quad \text{for } (t, \boldsymbol{x}) \in \mathcal{T} \times \Omega \\ u(0, \boldsymbol{x}; \boldsymbol{\mu}) &= u_0(\boldsymbol{x}; \boldsymbol{\mu}) \quad \text{for } \boldsymbol{x} \in \Omega \end{aligned} \tag{1}$$

where u_0 is the initial condition and f can include partial derivatives of u in $\boldsymbol{x}$. In the following, we always have appropriate boundary conditions so that the PDE problem (1) is well posed. The physics parameter $\boldsymbol{\mu}$ can enter the dynamics via f and the initial condition u_0. Standard numerical methods such as finite-element (Hughes, 2012) and finite-volume (LeVeque, 2002) methods can be used to numerically solve (1) to obtain a numerical solution $u_{\mathrm{F}}(\cdot, \cdot; \boldsymbol{\mu}) : \mathcal{T} \times \Omega \to \mathbb{R}$ for a physics parameter $\boldsymbol{\mu} \in \mathcal{D}$.

Computational procedures for learning reduced models (Antoulas, 2005; Rozza et al., 2008; Benner et al., 2015; Antoulas et al., 2021; Peherstorfer, 2022; Kramer et al., 2024) are typically split into an offline and an online phase: In the offline (training) phase, the reduced model is constructed from training trajectories

$$u_{\mathrm{F}}(\cdot, \cdot; \boldsymbol{\mu}_1), \ldots, u_{\mathrm{F}}(\cdot, \cdot; \boldsymbol{\mu}_m) : \Omega \times \mathcal{T} \to \mathbb{R} \tag{2}$$

over offline physics parameters $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_m \in \mathcal{D}$, which have been computed with the high-fidelity numerical model. In the subsequent online phase, the reduced model is used to rapidly predict solution fields at new physics parameters and initial conditions.

3. CoLoRA neural networks

We introduce CoLoRA networks that (a) provide nonlinear parameterizations that circumvent the Kolmogorov barrier of linear model reduction and (b) impose an inductive bias that treats the time variable t differently from the spatial variable $\boldsymbol{x}$, which reflects that time is a special variable in physics. In particular, CoLoRA networks allow a continuous adaptation of a low number of network weights over time to capture the dynamics of solution fields ("fine-tuning") for different physics parameters.

3.1. LoRA layers

CoLoRA networks are motivated by LoRA (Hu et al., 2022), a method that has been introduced to fine-tune large language models on discrete downstream tasks. LoRA layers are defined as

$$\mathcal{C}(\boldsymbol{x}) = \boldsymbol{W}\boldsymbol{x} + \Delta\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b} \tag{3}$$

where $\boldsymbol{x} \in \mathbb{R}^d$ is the input vector, $\boldsymbol{W}, \Delta\boldsymbol{W} \in \mathbb{R}^{n \times d}$ are weight matrices, and $\boldsymbol{b} \in \mathbb{R}^n$ is the bias term. The key of LoRA is that only $\Delta\boldsymbol{W}$ is changed during fine tuning and that $\Delta\boldsymbol{W}$ is of low rank $r \ll \min\{n, d\}$ so it can be parameterized as,

$$\Delta\boldsymbol{W} = \boldsymbol{A}\boldsymbol{B}\,, \qquad \boldsymbol{A} \in \mathbb{R}^{n \times r}, \boldsymbol{B} \in \mathbb{R}^{r \times d}\,.$$

Thus, only $n \times r + r \times d \ll n \times d$ parameters need to be update per layer during fine-tuning rather than all $n \times d + n$ as during pre-training,

3.2. The CoLoRA layer

Models with low intrinsic dimension are very common not only in large language models but also in many applications in science and engineering with phenomena that are described by PDEs (Bachmayr et al., 2017; Grasedyck et al., 2013; Berman & Peherstorfer, 2023). However, in the PDE settings, we have the special time variable t that requires us to "fine-tune" continuously as the PDE solution trajectories evolve; see Figure 1. Additionally, time imposes causality, which we want to preserve in CoLoRA models.

To enable a continuous low-rank adaptation, we introduce Continuous LoRA (CoLoRA) layers,

$$\mathcal{C}(\boldsymbol{x}) = \boldsymbol{W}\boldsymbol{x} + \alpha(t; \boldsymbol{\mu})\boldsymbol{A}\boldsymbol{B}\boldsymbol{x} + \boldsymbol{b} \tag{4}$$

where $\Delta\boldsymbol{W} = \boldsymbol{A}\boldsymbol{B}$ is a low-rank matrix of rank r that is trained offline and $\alpha(t; \boldsymbol{\mu}) \in \mathbb{R}$ is the online ("fine-tuning") parameter that can change continuously with t and also with the physics parameter $\boldsymbol{\mu}$ in the online phase of model reduction. For example, when using a multilayer perceptron (MLP) with L-many $\mathcal{C}$ layers and a linear output layer $\boldsymbol{c} \in \mathbb{R}^n$, we obtain

$$\hat{u}(\boldsymbol{x}; \boldsymbol{\theta}, \boldsymbol{\phi}(t, \boldsymbol{\mu})) = \boldsymbol{c}^T(\mathcal{C}_1(\sigma(\mathcal{C}_2(\ldots \sigma(\mathcal{C}_L(\boldsymbol{x}))\ldots)) \tag{5}$$

with activation function σ and $\mathcal{C}_i(\boldsymbol{x}) = \boldsymbol{W}_i\boldsymbol{x} + \alpha_i(t, \boldsymbol{\mu})\boldsymbol{A}_i\boldsymbol{B}_i\boldsymbol{x} + \boldsymbol{b}_i$ for $i = 1, \ldots, L$. The online parameters are given by the vector $\boldsymbol{\phi}(t, \boldsymbol{\mu}) = [\alpha_1(t, \boldsymbol{\mu}), \ldots, \alpha_q(t, \boldsymbol{\mu})] \in \mathbb{R}^q$ with $q = L$ in the example (5). We will later refer to $\boldsymbol{\phi}(t, \mu)$ as the latent state. All other CoLoRA parameters that are independent of time t and physics parameter $\boldsymbol{\mu}$ are trainable offline and collected into the offline parameter vector $\boldsymbol{\theta} \in \mathbb{R}^p$ of dimension $p \gg q$.

We note that in principle $\boldsymbol{A}\boldsymbol{B}$ could be full rank without increasing the size of q. But this would increase the number

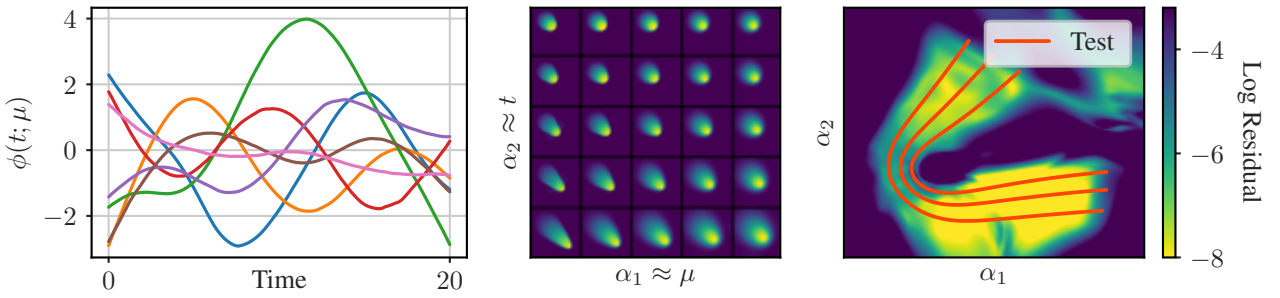

Figure 2. **Left:** Shows that CoLoRA's latent states $\boldsymbol{\phi}(t;\boldsymbol{\mu})$ adapt smoothly over time (RDE example). **Middle:** Training a CoLoRA model with a $q = 2$-dimensional latent state on the Burgers' example gives the first latent component corresponding to translation in time and the second one to the viscosity μ. **Right:** CoLoRA learns a continuous region of low PDE residual along which the latent trajectories evolve (Vlasov example); see Appendix B.

of parameters in $\boldsymbol{\theta}$. Additionally, the authors of LoRA (Hu et al., 2022) observed that full rank fine-tuning updates under-perform low rank ones despite having more degrees of freedom, which is also in agreement with the low ranks used in dynamic low-rank and online adaptive methods in model reduction (Koch & Lubich, 2007; Sapsis & Lermusiaux, 2009; Peherstorfer & Willcox, 2015; Einkemmer & Lubich, 2019; Peherstorfer, 2020; Uy et al., 2022; Singh et al., 2023).

A CoLoRA network defines a function $\hat{u} : \Omega \times \mathbb{R}^p \times \mathbb{R}^q \to \mathbb{R}$ that depends on an input $\boldsymbol{x} \in \Omega$, which is the spatial coordinate in our PDE setting, the offline parameters $\boldsymbol{\theta} \in \mathbb{R}^p$ that are independent of time t and physics parameter $\boldsymbol{\mu}$ and the online parameters or latent state $\boldsymbol{\phi}(t,\boldsymbol{\mu}) \in \mathbb{R}^q$ that depends on t and $\boldsymbol{\mu}$. A CoLoRA network can also output more than one quantity by modifying the output layer, which we will use for approximating systems of PDEs in the numerical experiments. A CoLoRA network $\hat{u}$ is an implicit neural representation (Sitzmann et al., 2020; Pan et al., 2023) in the sense that the PDE solution field is given implicitly by the parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}(t,\boldsymbol{\mu})$ and it can be evaluated at any coordinate $\boldsymbol{x} \in \Omega$, irrespective of discretizations and resolutions used during training.

If $\alpha_i(t;\boldsymbol{\mu})$ is a scalar for each layer $i = 1,\dots,L$, then the dimension of $\boldsymbol{\phi}(t,\boldsymbol{\mu})$ equals the number of layers in $\hat{u}$ in the MLP example (5), which can be overly restrictive. So we additional allow to have r-many online parameters $\alpha_1(t,\boldsymbol{\mu}),\dots,\alpha_r(t,\boldsymbol{\mu})$ for each $\mathcal{C}$ layer, in which case we have $\mathcal{C}(\boldsymbol{x}) = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{A}\,\mathrm{diag}(\alpha_1(t;\boldsymbol{\mu}),\dots,\alpha_r(t;\boldsymbol{\mu}))\boldsymbol{B}\boldsymbol{x} + \boldsymbol{b}$. The dimension of the online parameter vector $\boldsymbol{\phi}(t,\boldsymbol{\mu})$ is then $q = r \times L$. Other approaches are possible to make CoLoRA networks more expressive such as allowing $\boldsymbol{A}$ and $\boldsymbol{B}$ to depend on t and $\boldsymbol{\mu}$ as well; however, as we will show with our numerical experiments, only very few online parameters are needed in our experiments.

Other works have examined similar weight matrix decomposition to separate out a set of adaptable parameters in the context of PDEs (Cho et al., 2023; Wen et al., 2023) and in implicit neural representations (Kim et al., 2023); however, none directly treat the low rank adaptation as a function of time, which is key for our approach.

3.3. CoLoRA networks can circumvent the Kolmogorov barrier

CoLoRA networks can be nonlinear in the online parameter $\boldsymbol{\phi}(t,\boldsymbol{\mu})$ and thus achieve faster error decays than given by (linear) Kolmogorov n-widths. We give one example by considering the linear advection equation $\partial_t u(t,x) + \mu\partial_x u(t,x) = 0$ as in Ohlberger & Rave (2016) with initial condition $u(0,x) = u_0(x)$ and solution $u(x,t;\mu) = u_0(x - t\mu)$, which can lead to a slow n-width decay of $1/\sqrt{n}$ for a linear parameterization with n parameters. In contrast, with the CoLoRA MLP network $\hat{u}(x;\boldsymbol{\theta},\boldsymbol{\phi}(t,\mu)) = c(\mathcal{C}_1(\sigma_1(\mathcal{C}_2(\sigma_2(\mathcal{C}_3(x))))))$ with $L = 3$ layers, we can exactly represent translation and thus the solution $u(t,x;\mu) = u_0(x - t\mu)$ of the linear advection equation: Set $\boldsymbol{W}_3 = [1,0]^T, \boldsymbol{b}_3 = [0,1]^T, \alpha_3(t,\boldsymbol{\mu}) = 0$ and $\boldsymbol{A}_3, \boldsymbol{B}_3$ arbitrary. Further $\boldsymbol{W}_2 = [1,0], \boldsymbol{b}_2 = [0], \alpha_2(t,\boldsymbol{\mu}) = -t\boldsymbol{\mu}$ and $\boldsymbol{A}_2\boldsymbol{B}_2 = [0,1]$ and $\boldsymbol{W}_1 = [1], \boldsymbol{b}_1 = [0], \alpha_1(t,\boldsymbol{\mu}) = 0$. If we use the known initial condition as activation function $\sigma_1 = u_0$ and the identity as activation function σ_2 and set $c = 1$, then we obtain $\hat{u}(x,\boldsymbol{\theta},\boldsymbol{\phi}(t,\boldsymbol{\mu})) = u_0(x - t\mu)$, which is the solution of the linear advection example above. Note that using the initial condition as an activation function is proper in this context because the initial condition is typically given in closed form or at least can be evaluated over x and thus can be fitted during the pre-training.

Of course this example with the linear advection equation is contrived but it shows how translation can be represented well by CoLoRA networks, which is the challenge that leads to the Kolmogorov barrier (Peherstorfer, 2022). A more detailed treatment of the approximation theoretic properties of CoLoRA networks remains an open theory question that we leave for future work.

4. Training CoLoRA models offline

The goal of the following training procedure is to learn the offline parameters $\boldsymbol{\theta}$ of a CoLoRA network $\hat{u}$ for a given parameterized PDE (1) so that only the much lower dimensional latent state $\boldsymbol{\phi}(t, \boldsymbol{\mu})$ has to be updated online over time t and physics parameters $\boldsymbol{\mu}$ to approximate well the solution of the PDE.

Enforcing continuity in time CoLoRA models make a careful treatment of time t, which enters via the latent state $\boldsymbol{\phi}(t, \boldsymbol{\mu})$ rather than as input as the spatial coordinates $\boldsymbol{x}$; see also the discussion in Section 1. We want that special meaning of the time variable to be also reflected in the pre-training approach in the sense of imposing regularity in the latent state $\boldsymbol{\phi}(t, \boldsymbol{\mu})$ with respect to time t. Having smooth latent dynamics is a key property that has many desirable outcomes such as rapid time-stepping with large time step sizes, stability, and robustness to numerical perturbations. A naive pre-training of CoLoRA over $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ with a global optimization problem would allow $\boldsymbol{\phi}$ to change arbitrarily, i.e., in a non-smooth way, over time.

To impose regularity with respect to t, we introduce a hyper-network $h : \mathcal{T} \times \mathcal{D} \times \mathbb{R}^{q'} \to \mathbb{R}^q$ that depends on the parameter vector $\boldsymbol{\psi} \in \mathbb{R}^{q'}$; see (de Avila Belbute-Peres et al., 2021; Pan et al., 2023; Cho et al., 2023) for other methods that build on hyper-networks in different settings. Time t and physics parameter $\boldsymbol{\mu}$ are inputs to the hyper-network and $\boldsymbol{\phi}(t, \boldsymbol{\mu})$ is the output. We focus on the case where h is an MLP, but we stress that any other regression model can be used that provides the necessary regularity from t to $\boldsymbol{\phi}(t, \boldsymbol{\mu})$. For our purposes here, it is sufficient to choose h such that it is continuous in t. This means that its output written as $\boldsymbol{\phi}(t; \boldsymbol{\mu})$ will depend continuously on time t.

Loss function for pre-training Recall that we have access to training data in the form of solution fields for $\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_m$ given in (2). For $i = 1, \ldots, m$, we consider finite sets $\mathcal{X}_i \subset \Omega$ and $\mathcal{T}_i \subset \mathcal{T}$ of spatial coordinates $\boldsymbol{x}$ and time t samples over the spatial domain and time domain, respectively. For each offline physics parameter $\boldsymbol{\mu}_i$, we consider the relative error over the cross product $\mathcal{X}_i \times \mathcal{T}_i$

$$J_i(\boldsymbol{\theta}, \boldsymbol{\psi}) = \sum_{\substack{\boldsymbol{x} \in \mathcal{X}_i \\ t_i \in \mathcal{T}_i}} \frac{|u_{\mathrm{F}}(\boldsymbol{x}, t; \boldsymbol{\mu}_i) - \hat{u}(\boldsymbol{x}; \boldsymbol{\theta}, h(t, \boldsymbol{\mu}; \boldsymbol{\psi}))|^2}{|u_{\mathrm{F}}(\boldsymbol{x}, t; \boldsymbol{\mu}_i)|^2},$$

where u_{F} denotes the training trajectory for $\boldsymbol{\mu}_i$ that is available from the high-fidelity numerical model, $\hat{u}$ is our CoLoRA parameterizations, and h is the hyper-network discussed in the previous paragraph.

The loss that we optimize for the offline parameters $\boldsymbol{\theta} \in \mathbb{R}^p$ and the parameter vector $\boldsymbol{\psi} \in \mathbb{R}^{q'}$ of the hyper-network h averages the relative errors J_i over all training physics parameters,

$$L(\boldsymbol{\theta}, \boldsymbol{\psi}) = \frac{1}{m} \sum_{i=1}^{m} J_i(\boldsymbol{\theta}, \boldsymbol{\psi}). \tag{6}$$

This is also the mean relative error that we report on test parameters in our experiments; see Section 6.

We stress that in the spirit of implicit neural representations, the pre-training (as well as online) approach is independent of grids in the spatial and time domain. In fact, our formulation via the sets $\mathcal{X}_i$ and $\mathcal{T}_i$ for each training physics parameter $i = 1, \ldots, m$ allows for different, unstructured samples in the spatial and time domain for each training physics parameter.

5. Online phase of CoLoRA models

Given a new parameter $\boldsymbol{\mu}$ that we have not seen during training, the goal of the online phase is to rapidly approximate the high-fidelity numerical solution u_{F} at $\boldsymbol{\mu}$. With pre-trained CoLoRA models, we can go about this in two fundamentally different ways: First, we can take a purely data-driven route and simply evaluate the hyper-network h at the new $\boldsymbol{\mu}$ at any time t. Second, because the latent state $\boldsymbol{\phi}(t, \boldsymbol{\mu})$ depends on time t, we can take an equation-driven route and use the governing equation given in (1) to derive the online parameter $\boldsymbol{\phi}(t, \boldsymbol{\mu})$ via a variational formulation such as Neural Galerkin schemes (Bruna et al., 2024); see (Lasser & Lubich, 2020; Du & Zaki, 2021; Anderson & Farazmand, 2022; Berman & Peherstorfer, 2023) for other sequential-in-time methods that could be combined with CoLoRA.

5.1. Data-driven forecasting (CoLoRA-D)

We refer to CoLoRA models as CoLoRA-D if predictions at a new physics parameter $\boldsymbol{\mu}$ are obtained by evaluating the hyper-network h at $\boldsymbol{\mu}$ and the times t of interest. The predictions that are obtained from CoLoRA-D models are purely data-driven and therefore do not directly use the governing equations in any way; neither during the pre-training nor during the online phase. Reduced models based on CoLoRA-D are non-intrusive (Ghattas & Willcox, 2021; Kramer et al., 2024), which can have major advantages in terms of implementation and deployment because only data needs to be available; these advantages are the same as for operator learning (Li et al., 2021; Lu et al., 2021) that also is non-intrusive and typically relies only on data rather than the governing equations. The accuracy of CoLoRA-D models, however, critically depends on the generalization of h, which is in agreement with data-driven forecasting in general that has to rely on the generalization of data-fitted functions alone.

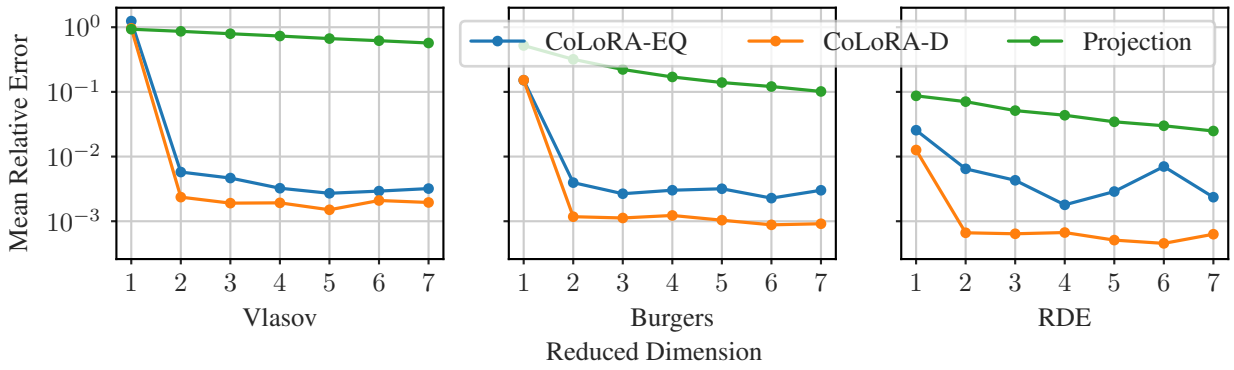

Figure 3. CoLoRA models achieve orders of magnitude lower errors than the best-approximation error of linear model reduction methods, which is in agreement with Section 3.3 that states that CoLoRA parameterizations circumvent the Kolmogorov barrier.

5.2. Equation-driven predictions (CoLoRA-EQ)

If the governing equations given in (1) are available, they can be used together with CoLoRA models to compute the states $\boldsymbol{\phi}(t, \boldsymbol{\mu})$ for a new parameter $\boldsymbol{\mu}$ in a variational sense. We follow Neural Galerkin schemes (Bruna et al., 2024), which provide a method for solving for parameters that enter non-linearly so that the corresponding parameterizations, in a variational sense, solve the given PDE. We stress that such a variational formulation is possible with CoLoRA models because the latent state $\boldsymbol{\phi}(t, \boldsymbol{\mu})$ depends on time t rather than time being an input as the spatial coordinate $\boldsymbol{x}$. In particular, the sequential-in-time training of Neural Galerkin schemes is compatible with the time-dependent online parameter $\boldsymbol{\phi}$. Together with Neural Galerkin schemes, CoLoRA provides solutions that are causal, which is different from many purely data-driven methods.

Neural Galerkin schemes build on the Dirac-Frenkel variational principle (Dirac, 1930; Frenkel, 1934; Lubich, 2008; Lasser & Lubich, 2020), which can be interpreted as finding time derivatives $\dot{\boldsymbol{\phi}}(t, \boldsymbol{\mu})$ that solve the Galerkin condition

$$\langle \partial_{\phi_i} \hat{u}, r_t(\boldsymbol{\phi}(t, \boldsymbol{\mu}), \dot{\boldsymbol{\phi}}(t, \boldsymbol{\mu}) \rangle = 0 \,, \quad i = 1, \ldots, q \,,$$

so that the residual $r_t(\boldsymbol{\phi}, \dot{\boldsymbol{\phi}}) = \partial_t \hat{u} - f(t, \cdot, \hat{u})$ of the PDE (1) as a function over the spatial domain Ω is orthogonal to the tangent space of the manifold $\{\hat{u}(\cdot; \boldsymbol{\theta}, \boldsymbol{\phi}) \,|\, \boldsymbol{\phi} \in \mathbb{R}^q\}$ induced by the online parameters at the current function $\hat{u}(\cdot; \boldsymbol{\theta}, \boldsymbol{\phi})$; we refer to (Bruna et al., 2024) for details and to Appendix H for the computational procedure.

The key feature of the equation-driven approach for predictions with CoLoRA models is that the latent states are optimal in a Galerkin sense, which provides a variational interpretation of the solution $\hat{u}$ and opens the door to using residual-based error estimators to provide accuracy guarantees, besides other theory tools. Additionally, as mentioned above, it imposes causality, which is a fundamental principle in science that we often want to preserve in numerical simulations. Furthermore, using the governing equations is helpful to conserve quantities such as energy, mass, momentum, which we will demonstrate in Section 6.6.

6. Numerical experiments

6.1. PDE problems

The following three problems are challenging to reduce with conventional linear model reduction methods because the dynamics are transport dominated (Peherstorfer, 2022). Additional details on these equations and the full order models are provided in Appendix C.

Collisionless charged particles in electric field The Vlasov equation describes the motion of collisionless charged particles under the influence of an electric field. We consider the setup of Güçlü et al. (2014), which demonstrates filamentation of the distribution function of charged particles as they are affected by the electric field. Our physics parameter $\mu \in [0.2, 0.4]$ enters via the initial condition. The full numerical model benchmarked in Figure 4 uses second-order finite differences on a 1024×1024 grid with adaptive time integrator.

Burgers' equation in 2D Fields governed by the Burgers' equations can form sharp advecting fronts. The sharpness of these fronts are controlled by the viscosity parameter which we use as our physics parameter $\mu \in [10^{-3}, 10^{-2}]$. The full model benchmarked in Figure 4 uses 2nd-order finite differences and a 1024×1024 spatial grid with an implicit time integration scheme using a time step size of $1\mathrm{e}{-3}$.

Rotating denotation waves We consider a model of rotating detonation waves, which is motivated by space propulsion with rotating detonation engines (RDE) (Koch et al., 2020; Anand & Gutmark, 2019; Raman et al., 2023). The physics parameter μ we reduce over corresponds to the combustion injection rate, which leads to bifurcation phenomena that we investigate over the interval $\mu \in [2.0, 3.1]$. The full model benchmarked in Figure 4 uses a finite volume method on a 2048 grid with an implicit time integration scheme using a time step size of $1\mathrm{e}{-3}$.

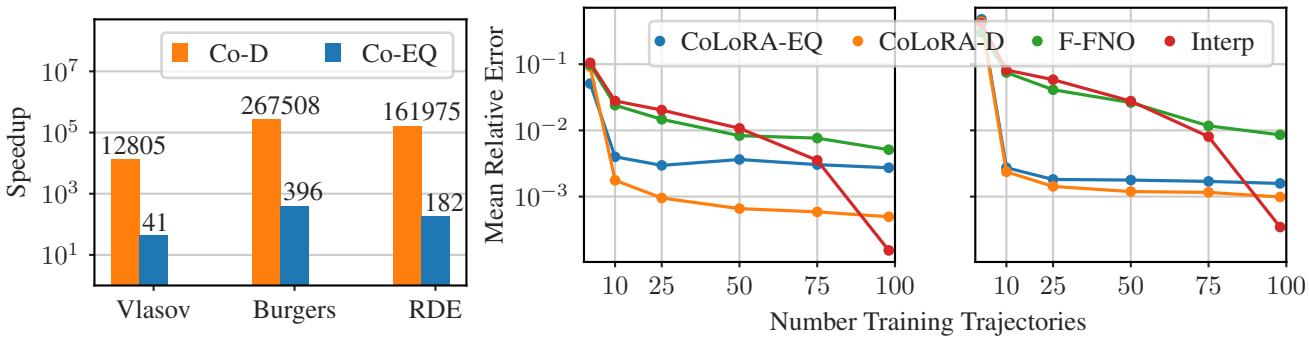

Figure 4. **Left**: Purely data-driven CoLoRA (CoLoRA-D) is more than four orders of magnitude faster than traditional numerical models. If the governing equations are solved with Neural Galerkin (Bruna et al., 2024) in a Galerkin-optimal variational sense in the CoLoRA parameterization (CoLoRA-EQ), we still obtain about two orders of magnitude speedups while maintaining causality in the solution. **Right**: CoLoRA is more data efficient than operator learning and thus well suited for low-data regimes (Burgers', Vlasov).

Other PDE models We also look at other PDEs to benchmark against methods from the literature; see Table 1. These include a two-dimensional wave problem with a four-dimensional physics parameter taken from Yin et al. (2023) and a three-dimensional shallow water wave example from Serrano et al. (2023).

6.2. CoLoRA architectures

The reduced-model parameterization $\hat{u}$ is an MLP with CoLoRA layers. The hyper-network h is an MLP with regular linear layers. Both use swish activation functions. The most important architectural choice we make is the size of our networks—$\hat{u}$ has 8 layers each of width 25 and h has three layers each of width 15. As discussed earlier, such small networks are sufficient because of the strong inductive bias and low-rank continuity in time of CoLoRA networks. Only for the 3D shallow water example we use layer width 128. The larger width helps to capture the oscillations in the solution field in this example; see also Section 7. The error metric we report is the mean relative error, which is also our loss function (6). More details are in Appendix D and Figure 7.

6.3. CoLoRA and number of latent parameters

Figure 3 compares the mean relative error of the proposed CoLoRA models with conventional linear projections, which serve as the empirical best-approximation error that can be achieved with any linear model reduction method (see Appendix F). In all examples, the error is shown for test physics parameters that have not been used during training; see Appendix G.

First, the linear approximations are ineffective for all three examples, which is in agreement with the observation from Section 6.1 that the three PDE models are challenging to reduce with linear model reduction methods. Second, our CoLoRA models achieve orders of magnitude lower relative errors for the same number of parameters as linear approximations. In all examples, 2–3 latent parameters are sufficient in CoLoRA models, which is in agreement with the low dimensionality of the physics parameters of these models. After the steep drop off of the error until around $q = 2$ online parameters, there is a slow improvement if any as we increase q, which is in agreement with other nonlinear approximations methods (Chen et al., 2023b; Lee & Carlberg, 2020). This is because once q is equal to the intrinsic dimension of the problem, compression no longer helps reduce errors in predictions and instead the error is driven by other error sources such as time integration and generalization of the hyper-network. In these examples, the purely data-driven CoLoRA-D achieves slightly lower relative errors than the equation-driven CoLoRA-EQ, which could be due to the time integration error. In any case, the CoLoRA-EQ results show that we learn representations that are consistent with the PDE dynamics in the sense of Neural Galerkin based on the Dirac-Frenkel variational principle.

6.4. Speedups of CoLoRA

In Figure 4, we show the relative speedup of CoLoRA when compared to the runtime of the high-fidelity numerical models based on finite-difference and finite-volume methods as described in Appendix C. The speedups in the Burgers' and the RDE examples are higher than in the Vlasov example because we use an explicit time integration scheme for Vlasov but implicit ones for Burgers' and RDE. When integrating the governing equations in CoLoRA-EQ, we achieve speedups because of the smoothness of the latent dynamics of $\phi(t;\mu)$ as shown in Figure 2. The smoothness allows us to integrate $\phi(t;\mu)$ with a solver that uses an adaptive time-step control, which adaptively selects large time steps due to the smoothness of the dynamics. When using CoLoRA-D, we achieve orders of magnitude higher speedups because forecasting requires evaluating the hyper-network h only. This can be done quickly due to the small size of the hyper-network h as described in Section 6.2. We

note that we benchmark our method on the time it takes to compute the latent state $\phi(t;\mu)$ on the same time grid as the full model. There will of course be additional computational costs associated with plotting the CoLoRA solution on a grid in Ω.

6.5. Data efficiency versus operator learning

A key difference to operator learning is that CoLoRA aims to predict well the influence of the physics parameter $\boldsymbol{\mu}$ on the solution fields, rather than aiming for a generic operator that maps a solution at one time step to the next. We now show that CoLoRA can leverage the more restrictive problem formulation so that fewer training trajectories are sufficient. As Figure 4 shows for the Burgers' and Vlasov example, we achieve relative errors in the range of $1\mathrm{e}{-3}$ with only about $m = 10$ training trajectories, whereas the operator- learning variant F-FNO (Tran et al., 2023) based on Fourier neural operators (FNOs) (Li et al., 2021) leads to an about one order of magnitude higher relative error. Neural operators struggle to achieve relative errors below $1\mathrm{e}{-2}$, while CoLoRA achieves one order of magnitude lower relative errors with one order of magnitude fewer training trajectories. We also compare to simply linearly interpolating the function u_{F} over space, time, and parameter and observe that in low data regimes CoLoRA achieves orders of magnitude more accurate predictions. In high data regimes, for sufficiently smooth problems, linear interpolation becomes accurate as training physics parameters start to be closer and closer to test physics parameters; see Appendix G for details.

6.6. Leveraging physics knowledge in the online phase with CoLoRA-EQ

In the numerical experiments conducted so far, the purely data-driven CoLoRA-D outperforms CoLoRA-EQ in terms of error and speedup. However, using the physics equations online can be beneficial in other ways such as for causality and theoretical implications, especially for residual-based error estimators; see Section 5. We now discuss another one here numerically, namely conserving quantities during time integration. We build on conserving Neural Galerkin schemes introduced in Schwerdtner et al. (2023) to conserve the mass of the probability distribution that describes the particles in the Vlasov problem. Preserving unit mass can be important for physics interpretations. In Figure 5, we show that using the CoLoRA-EQ with Schwerdtner et al. (2023), we are able to conserve the mass of solution fields of the Vlasov equation to machine precision. By contrast, neither the CoLoRA-D nor F-FNO conserve the quantity, as the numerical results indicate.

6.7. Comparison to other nonlinear methods

We run CoLoRA on two benchmark problems. The first is described in the DINo publication (Yin et al., 2023). This is a 2D wave problem where the four dimensional parameter μ affects the position and magnitude of the initial condition. The second is described in the CORAL publication (Serrano et al., 2023). It is a shallow water equation formulated over a 3D spherical domain where the μ parameter nonlinearly affects the initial condition. We additionally report the accuracy of two other methods MP-PDE (Brandstetter et al., 2022) and DeepONet (Lu et al., 2021) both originally benchmarked in Yin et al. (2023) and Serrano et al. (2023). We report these results in Table 1. We see that on these two challenging benchmark problems CoLoRA achieves the lowest mean squared error. In terms of implementations, CoLoRA succeeds using a relatively simple modulation scheme and straightforward pre-training. CoLoRA also outperforms all other methods while using close to one to two orders of magnitude fewer parameters.

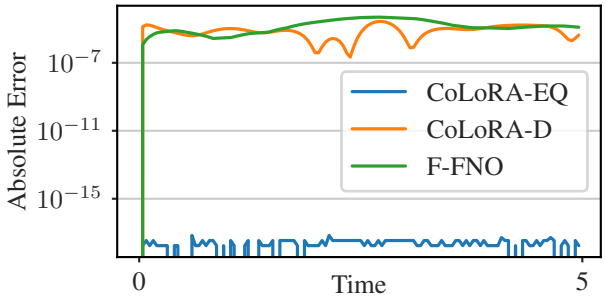

Figure 5. Solving the governing equations in a variational sense with Neural Galerkin (Bruna et al., 2024) and CoLoRA parameterizations (CoLoRA-EQ) leads to causal solutions and allows conserving quantities (Schwerdtner et al., 2023) such as mass in the Vlasov problem, which is key for building trust in physics predictions and for interpretability.

7. Conclusions, limitations, and future work

CoLoRA leverages that PDE dynamics are typically continuous in time while evolving on low-dimensional manifolds. CoLoRA models provide nonlinear approximations and therefore are efficient in reducing transport-dominated problems that are affected by the Kolmogorov barrier. At the same time, CoLoRA is data efficient and requires only few training trajectories in our examples. The continuous-in-time adaptation of CoLoRA network weights leads to rapid predictions of solution fields of PDEs at new physics parameters, which outperforms current state-of-the-art methods.

Limitations First, the theoretical analysis for reduced models based on CoLoRA is currently limited. The preliminary results on overcoming the Kolmogorov barrier for a specific setup with the linear advection equation cannot be directly generalized to other problems and thus a more in-depth analysis is necessary. Second, there are applications where pre-training the CoLoRA network once and for all

example:	**three-dim. spherical shallow water**		**two-dim. wave**	
metric:	MSE	number parameters	MSE	number parameters
MP-PDE	9.37e-5	-	9.256e-7	-
DeepONet	6.54e-3	-	1.847e-2	-
DINo	4.48e-5	2,022,912	9.495e-6	579,776
CORAL	3.44e-6	1,049,344	-	-
CoLoRA-D	**3.19e-06**	**335,744**	**1.891e-07**	**7505**

Table 1. CoLoRA is more accurate than a range of other methods for forecasting PDEs and model reduction based on implicit neural representations while using significantly fewer parameters. MSE values and parameter counts are taken from Yin et al. (2023) for the 2D wave problem and from Serrano et al. (2023) for 3D spherical shallow water. Parameter counts are estimated from the descriptions architecture depth and width in the original papers.

is insufficient, such as when predicting bifurcations that are not represented in the training data. Then, an online adaptive updating of the offline parameters is desired, for which efficient methods need to be developed.

Future work First, to well approximate solution fields with high-frequency oscillations, sharp gradients, and other non-smooth features, reduced modeling with CoLoRA can be combined with Fourier feature embeddings and periodic activation functions. Second, our hyper-network based method of modulation succeeds mainly when generalizing to examples which are in-distribution in terms of $\boldsymbol{\mu}$ and t. Later work might seek to expand CoLoRA's method of parameter modulation to settings with neural ordinary differential equations and other methods that can enhance CoLoRA's extrapolation ability. Third, a future direction is scaling reduced modeling with CoLoRA to higher-dimensional problems in both parameter and spatial domain. We expect that active data collection will be key for CoLoRA models to be efficient in high dimension.

We provide an implementation of CoLoRA at `https://github.com/julesberman/CoLoRA`.

Acknowledgements

The authors were partially supported by the National Science Foundation under Grant No. 2046521 and the Office of Naval Research under award N00014-22-1-2728. This work was also supported in part through the NYU IT High Performance Computing resources, services, and staff expertise.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Aghajanyan, A., Gupta, S., and Zettlemoyer, L. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 7319–7328, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.568. URL https://aclanthology.org/2021.acl-long.568.

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

Anand, V. and Gutmark, E. Rotating detonation combustors and their similarities to rocket instabilities. *Progress in Energy and Combustion Science*, 73:182–234, 2019. ISSN 0360-1285. doi: https://doi.org/10.1016/j.pecs.2019.04.001. URL https://www.sciencedirect.com/science/article/pii/S0360128518301783.

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

Antoulas, A. C. *Approximation of large-scale dynamical systems*. SIAM, 2005.

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

Bachmayr, M., Cohen, A., and Dahmen, W. Parametric PDEs: sparse or low-rank approximations? *IMA Journal*

of Numerical Analysis, 38(4):1661–1708, 09 2017. ISSN 0272-4979. doi: 10.1093/imanum/drx052. URL https://doi.org/10.1093/imanum/drx052.

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

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

Berman, J. and Peherstorfer, B. Randomized sparse Neural Galerkin schemes for solving evolution equations with deep networks. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=JTKd7zYROf.

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

Black, F., Schulze, P., and Unger, B. Projection-based model reduction with dynamically transformed modes. *ESAIM: M2AN*, 54(6):2011–2043, 2020.

Boullé, N. and Townsend, A. A Mathematical Guide to Operator Learning, December 2023. URL http://arxiv.org/abs/2312.14688. arXiv:2312.14688 [cs, math].

Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

Brandstetter, J., Worrall, D. E., and Welling, M. Message passing neural PDE solvers. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=vSix3HPYKSU.

Bruna, J., Peherstorfer, B., and Vanden-Eijnden, E. Neural Galerkin schemes with active learning for high-dimensional evolution equations. *Journal of Computational Physics*, 496:112588, January 2024. ISSN 0021-9991. doi: 10.1016/j.jcp.2023.112588. URL https://www.sciencedirect.com/science/article/pii/S0021999123006836.

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

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

Chen, H., Wu, R., Grinspun, E., Zheng, C., and Chen, P. Y. Implicit neural spatial representations for time-dependent PDEs. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 5162–5177. PMLR, 23–29 Jul 2023a. URL https://proceedings.mlr.press/v202/chen23af.html.

Chen, P. Y., Xiang, J., Cho, D. H., Chang, Y., Pershing, G. A., Maia, H. T., Chiaramonte, M. M., Carlberg, K. T., and Grinspun, E. CROM: Continuous reduced-order modeling of PDEs using implicit neural representations. In *The Eleventh International Conference on Learning Representations*, 2023b. URL https://openreview.net/forum?id=FUORz1tG8Og.

Cho, W., Lee, K., Rim, D., and Park, N. Hypernetwork-based meta-learning for low-rank physics-informed neural networks. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=dzqKAM2sKa.

Cohen, A. and DeVore, R. Kolmogorov widths under holomorphic mappings. *IMA J. Numer. Anal.*, 36(1):1–12, 2016. ISSN 0272-4979.

de Avila Belbute-Peres, F., fan Chen, Y., and Sha, F. HyperPINN: Learning parameterized differential equations with physics-informed hypernetworks. In *The Symbiosis of Deep Learning and Differential Equations*, 2021. URL https://openreview.net/forum?id=LxUuRDUhRjM.

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

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

Du, Y. and Zaki, T. A. Evolutional deep neural network. *Physical Review E*, 104(4), October 2021. ISSN 2470-0053. doi: 10.1103/physreve.104.045303. URL http://dx.doi.org/10.1103/PhysRevE.104.045303.

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

Ehrlacher, V., Lombardi, D., Mula, O., and Vialard, F.-X. Nonlinear model reduction on metric spaces. Application to one-dimensional conservative PDEs in Wasserstein spaces. *ESAIM Math. Model. Numer. Anal.*, 54(6):2159–2197, 2020. ISSN 0764-583X.

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

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

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/finn17a.html.

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

Fulton, L., Modi, V., Duvenaud, D., Levin, D. I. W., and Jacobson, A. Latent-space dynamics for reduced deformable simulation. *Computer Graphics Forum*, 2019.

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

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

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

Ghattas, O. and Willcox, K. Learning physics-based models from data: perspectives from inverse problems and model reduction. *Acta Numerica*, 30:445–554, 2021. doi: 10.1017/S0962492921000064.

Grasedyck, L., Kressner, D., and Tobler, C. A literature survey of low-rank tensor approximation techniques. *GAMM-Mitteilungen*, 36(1):53–78, 2013. doi: https://doi.org/10.1002/gamm.201310004. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/gamm.201310004.

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

Güçlü, Y., Christlieb, A. J., and Hitchon, W. N. Arbitrarily high order convected scheme solution of the vlasov–poisson system. *Journal of Computational Physics*, 270:711–752, August 2014. ISSN 0021-9991. doi: 10.1016/j.jcp.2014.04.003. URL http://dx.doi.org/10.1016/j.jcp.2014.04.003.

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

Hu, E. J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

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

Hughes, T. J. R. *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*. Dover Publications, 2012.

Iollo, A. and Lombardi, D. Advection modes by optimal mass transfer. *Phys. Rev. E*, 89:022923, Feb 2014.

Issan, O. and Kramer, B. Predicting solar wind streams from the inner-heliosphere to earth via shifted operator inference. *Journal of Computational Physics*, 473:111689, 2023. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2022.111689. URL https://www.sciencedirect.com/science/article/pii/S0021999122007525.

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

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

Khodak, M., Tenenholtz, N. A., Mackey, L., and Fusi, N. Initialization and regularization of factorized neural layers. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=KTlJT1nof6d.

Kim, C., Lee, D., Kim, S., Cho, M., and Han, W.-S. Generalizable implicit neural representations via instance pattern composers. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11808–11817, 2023. doi: 10.1109/CVPR52729.2023.01136.

Kim, Y., Choi, Y., Widemann, D., and Zohdi, T. A fast and accurate physics-informed neural network reduced order model with shallow masked autoencoder. *Journal of Computational Physics*, 451:110841, 2022. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2021.110841. URL https://www.sciencedirect.com/science/article/pii/S0021999121007361.

Kingma, D. and Ba, J. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA, 2015.

Koch, J., Kurosaka, M., Knowlen, C., and Kutz, J. N. Mode-locked rotating detonation waves: Experiments and a model equation. *Physical Review E*, 101(1), January 2020. ISSN 2470-0053. doi: 10.1103/physreve.101.013106. URL http://dx.doi.org/10.1103/PhysRevE.101.013106.

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

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

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

Lee, K. and Carlberg, K. T. Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders. *Journal of Computational Physics*, 404:108973, 2020. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2019.108973. URL https://www.sciencedirect.com/science/article/pii/S0021999119306783.

Lee, K. and Carlberg, K. T. Deep conservation: A latent-dynamics model for exact satisfaction of physical conservation laws. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(1):277–285, May 2021. doi: 10.1609/aaai.v35i1.16102. URL https://ojs.aaai.org/index.php/AAAI/article/view/16102.

Lee, K. and Parish, E. J. Parameterized neural ordinary differential equations: applications to computational physics problems. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 477 (2253):20210162, 2021. doi: 10.1098/rspa.2021.0162. URL https://royalsocietypublishing.org/doi/abs/10.1098/rspa.2021.0162.

LeVeque, R. J. *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2002.

Li, C., Farkhoor, H., Liu, R., and Yosinski, J. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=ryup8-WCW.

Li, Z., Kovachki, N. B., Azizzadenesheli, K., liu, B., Bhattacharya, K., Stuart, A., and Anandkumar, A. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=c8P9NQVtmnO.

Lu, L., Jin, P., Pang, G., Zhang, Z., and Karniadakis, G. E. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3):218–229, Mar 2021. ISSN 2522-5839. doi: 10.1038/s42256-021-00302-5. URL https://doi.org/10.1038/s42256-021-00302-5.

Lubich, C. *From quantum to classical molecular dynamics: reduced models and numerical analysis*, volume 12. European Mathematical Society, 2008.

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

Maday, Y., Patera, A. T., and Turinici, G. Global a priori convergence theory for reduced-basis approximations of single-parameter symmetric coercive elliptic partial differential equations. *C. R. Math. Acad. Sci. Paris*, 335(3):289–294, 2002. ISSN 1631-073X.

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

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

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

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

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

Pan, S., Brunton, S. L., and Kutz, J. N. Neural implicit flow: a mesh-agnostic dimensionality reduction paradigm of spatio-temporal data. *Journal of Machine Learning Research*, 24(41):1–60, 2023. URL http://jmlr.org/papers/v24/22-0365.html.

Papapicco, D., Demo, N., Girfoglio, M., Stabile, G., and Rozza, G. 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, 2022.

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

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

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

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

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

Raissi, M., Perdikaris, P., and Karniadakis, G. 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, 2019.

Raman, V., Prakash, S., and Gamba, M. Nonidealities in rotating detonation engines. *Annual Review of Fluid Mechanics*, 55(1):639–674, 2023. doi: 10.1146/annurev-fluid-120720-032612. URL https://doi.org/10.1146/annurev-fluid-120720-032612.

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

Reiss, J., Schulze, P., Sesterhenn, J., and Mehrmann, V. The shifted proper orthogonal decomposition: a mode decomposition for multiple transport phenomena. *SIAM J. Sci. Comput.*, 40(3):A1322–A1344, 2018. ISSN 1064-8275.

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

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

Rozza, G., Huynh, D. B. P., and Patera, A. T. 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, 2008.

Sainath, T. N., Kingsbury, B., Sindhwani, V., Arisoy, E., and Ramabhadran, B. Low-rank matrix factorization for deep neural network training with high-dimensional output targets. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 6655–6659, 2013. doi: 10.1109/ICASSP.2013.6638949.

Sapsis, T. P. and Lermusiaux, P. F. Dynamically orthogonal field equations for continuous stochastic dynamical systems. *Physica D: Nonlinear Phenomena*, 238(23): 2347–2360, 2009. ISSN 0167-2789.

Schotthöfer, S., Zangrando, E., Kusch, J., Ceruti, G., and Tudisco, F. Low-rank lottery tickets: finding efficient low-rank neural networks via matrix differential equations. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A. (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 20051–20063. Curran Associates, Inc., 2022.

Schwerdtner, P., Schulze, P., Berman, J., and Peherstorfer, B. Nonlinear embeddings for conserving Hamiltonians and other quantities with Neural Galerkin schemes, October 2023. URL http://arxiv.org/abs/2310.07485. arXiv:2310.07485 [cs, math].

Serrano, L., Boudec, L. L., Koupaï, A. K., Wang, T. X., Yin, Y., Vittaut, J.-N., and Gallinari, P. Operator learning with neural fields: Tackling PDEs on general geometries. In *Thirty-seventh Conference on Neural In-*

formation Processing Systems, 2023. URL https://openreview.net/forum?id=4jEjq5nhg1.

Singh, R., Uy, W., and Peherstorfer, B. Lookahead data-gathering strategies for online adaptive model reduction of transport-dominated problems. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 2023.

Sitzmann, V., Martel, J., Bergman, A., Lindell, D., and Wetzstein, G. Implicit neural representations with periodic activation functions. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 7462–7473. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/53c04118df112c13a8c34b38343b9c10-Paper.pdf.

Taddei, T., Perotto, S., and Quarteroni, A. Reduced basis techniques for nonlinear conservation laws. *ESAIM Math. Model. Numer. Anal.*, 49(3):787–814, 2015. ISSN 0764-583X.

Tran, A., Mathews, A., Xie, L., and Ong, C. S. Factorized Fourier Neural Operators, March 2023. URL http://arxiv.org/abs/2111.13802. arXiv:2111.13802 [cs].

Uy, W. I. T., Wentland, C. R., Huang, C., and Peherstorfer, B. Reduced models with nonlinear approximations of latent dynamics for model premixed flame problems. *arXiv*, 2209.06957, 2022.

Wan, Z. Y., Zepeda-Nunez, L., Boral, A., and Sha, F. Evolve smoothly, fit consistently: Learning smooth latent dynamics for advection-dominated systems. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Z4s73sJYQM.

Wang, Y., Navon, I. M., Wang, X., and Cheng, Y. 2d Burgers equation with large Reynolds number using POD/DEIM and calibration. *International Journal for Numerical Methods in Fluids*, 82 (12):909–931, 2016. doi: https://doi.org/10.1002/fld.4249. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.4249.

Wen, T., Lee, K., and Choi, Y. Reduced-order modeling for parameterized PDEs via implicit neural representations, November 2023. URL http://arxiv.org/abs/2311.16410. arXiv:2311.16410 [math-ph, physics:physics].

Wen, Y., Vanden-Eijnden, E., and Peherstorfer, B. Coupling parameter and particle dynamics for adaptive sampling in Neural Galerkin schemes. *Physica D: Nonlinear Phenomena*, 462:134129, 2024.

Yin, Y., Kirchmeyer, M., Franceschi, J.-Y., Rakotomamonjy, A., and Gallinari, P. Continuous PDE Dynamics Forecasting with Implicit Neural Representations, February 2023. URL http://arxiv.org/abs/2209.14855. arXiv:2209.14855 [cs, stat].

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

Zhang, Y., Chuangsuwanich, E., and Glass, J. Extracting deep neural network bottleneck features using low-rank matrix factorization. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 185–189, 2014. doi: 10.1109/ICASSP.2014.6853583.

Zhao, Y., Li, J., and Gong, Y. Low-rank plus diagonal adaptation for deep neural networks. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5005–5009, 2016. doi: 10.1109/ICASSP.2016.7472630.

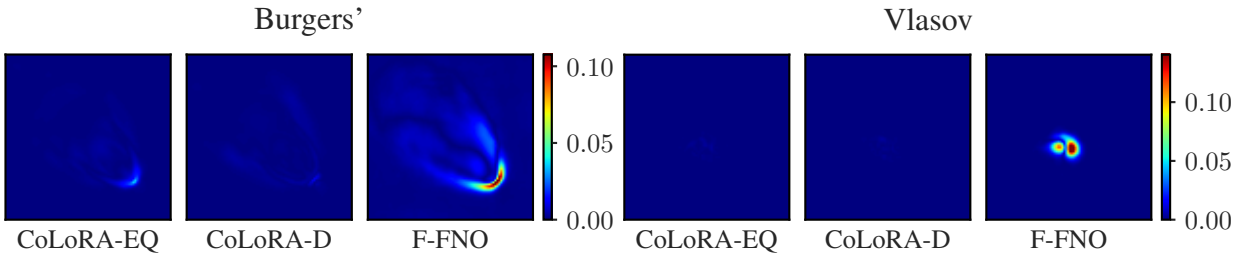

Figure 6. We show the point-wise absolute error of CoLoRA vs F-FNO. CoLoRA provides accurate solution fields even when trained on a low number of trajectories compared to operator learning. Plots here shown at 10 training trajectories.

A. Literature review of nonlinear model reduction

There is a wide range of literature on model reduction; see Antoulas (2005); Rozza et al. (2008); Benner et al. (2015); Antoulas et al. (2021); Kramer et al. (2024) for surveys and textbooks. We focus here on model reduction methods that build on nonlinear parameterizations to circumvent the Kolmogorov barrier (Peherstorfer, 2022).

First, there is a range of methods that pre-compute a dictionary of basis functions and then subselect from the dictionary in the online phase (Jens L. Eftang & Patera, 2011; Dihlmann et al., 2011; Amsallem et al., 2012; Eftang & Stamm, 2012; Maday & Stamm, 2013; Peherstorfer et al., 2014; Kaulmann et al., 2015; Geelen & Willcox, 2022). However, once the dictionary has been pre-computed offline, it remains fixed and thus such dictionary-based localized model reduction methods are less flexible in this sense compared to the proposed CoLoRA approach.

Second, there are nonlinear reduced modeling methods that build on nonlinear transformations to either recover linear low-rank structure that can be well approximated with linear parameterizations in subspace or that augment linear approximations with nonlinear correction terms. For example, the early work Rowley & Marsden (2000) shows how to shift bases to account for translations and other symmetries. Other analytic transformations are considered in, e.g., Ohlberger & Rave (2013); Reiss et al. (2018); Ehrlacher et al. (2020); Qian et al. (2020); Papapicco et al. (2022); Barnett & Farhat (2022); Geelen et al. (2023); Issan & Kramer (2023). The works by Taddei et al. (2015); Cagniart et al. (2019) parameterize the transformation maps and train their parameters on snapshot data rather than using transformations that are analytically available.

Third, there are online adaptive model reduction methods that adapt the basis representation during the online phase (Koch & Lubich, 2007; Sapsis & Lermusiaux, 2009; Iollo & Lombardi, 2014; Gerbeau & Lombardi, 2014; Carlberg, 2015; Peherstorfer & Willcox, 2015; Zahr & Farhat, 2015; Peherstorfer, 2020; Black et al., 2020; Billaud-Friess & Nouy, 2017; Ramezanian et al., 2021; Huang & Duraisamy, 2023). An influential line of work is the one on dynamic low-rank approximations (Koch & Lubich, 2007; Musharbash et al., 2015; Einkemmer & Lubich, 2019; Einkemmer et al., 2021; Musharbash et al., 2015; Musharbash & Nobile, 2017; 2018; Hesthaven et al., 2022) that adapt basis functions with low-rank additive updates over time and thus can be seen as a one-layer version of CoLoRA reduced models.

B. Details on numerical experiments

In Figure 2 we have three plots which show the benefits of the CoLoRA method. In the left plot, we show each dimension of ϕ as a function of time. These parameters were generated through time integration (CoLoRA-EQ). This shows even with integration we get smooth dynamics. In the middle plot, we traverse the latent space by generating samples in the two dimensional space spanned by the component functions of $\phi(t;\boldsymbol{\mu})$ and then evaluating $\hat{u}$ at each of these points. In the right plot, we train CoLoRA on Vlasov with a reduced dimension of 2. We then show the magnitude of the PDE residual at grid of points in the two dimensional space spanned by the component functions of the learned $\phi(t;\boldsymbol{\mu})$. The magnitude of the PDE residual is given by the residual from solving the least squares problem given in (7) at each of these grid points. When plotting the resulting field we see that CoLoRA learns a continuous region of low PDE residual along which the latent training trajectories lie. The inferred latent test trajectories lie in between training trajectories showing the generalization properties of CoLoRA which allows for an accurate time continuous representation of the solution.

In Section 6.5 we report on the error of CoLoRA and F-FNO as a function of the number of training trajectories. In Figure 6, we give the point-wise error plots at 10 training trajectories. In particular we see that F-FNO has difficulty tracking the advection dynamics of the solution over time. CoLoRA by contrast is able to approximate these dynamics accurately.

All numerical experiments were implemented in Python with JAX (Bradbury et al., 2018) with just-in-time compilation enabled. All benchmarks were run on a single NVIDIA RTX-8000 GPU.

C. Description of full order models (FOMs)

In order to ensure a fair comparison in terms of runtime between CoLoRA and the FOMs, we implement all FOMs in JAX (Bradbury et al., 2018) with just-in-time compilation.

C.1. Vlasov

The Vlasov equations are

$$\partial_t u(t, \boldsymbol{x}; \mu) = -x_2 \partial_{x_1} u(t, \boldsymbol{x}; \mu) + \partial_{x_1} \phi(x_1; \mu) \partial_{x_2} u(t, \boldsymbol{x}; \mu)$$

where $\boldsymbol{x} = [x_1, x_2]^T \in \mathbb{R}^2$. The first coordinate x_1 corresponds to the position of the particles and x_2 to the velocity. The potential of the electric field is $\phi(x) = -(0.2 + 0.2\cos(\pi x^4) + 0.1\sin(\pi x))$. We impose periodic boundary conditions on $\mathcal{X} = [-1, 1)^2$ and solve over the time domain $\mathcal{T} = [0, 5]$. Our physics parameter $\mu \in [0.2, 0.4]$ enters via the initial condition $u_0(\boldsymbol{x}; \mu) = \exp(-100|(\boldsymbol{x} - 0.2 + \mu)|^2)$.

The Vlasov full order model uses a 4th order central difference stencil to compute spatial derivatives over a 1024×1024 spatial grid. This is then integrated using 5th order explicit Runge-Kutta method with an embedded 4th order method for adaptive step sizing.

C.2. Burgers'

The two-dimensional Burgers' equations are described as,

$$\begin{aligned}
\frac{\partial u}{\partial t} &= -u\frac{\partial u}{\partial x} - v\frac{\partial u}{\partial y} + \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) \\
\frac{\partial v}{\partial t} &= -u\frac{\partial v}{\partial x} - v\frac{\partial v}{\partial y} + \mu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right).
\end{aligned}$$

We consider the spatial domain $\mathcal{X} = [0, 1)^2$, time domain $\mathcal{T} = (0, 1]$ where $\boldsymbol{x} = [x, y]^T \in \mathbb{R}^2$ and the physics parameter $\mu \in [10^{-3}, 10^{-2}]$ corresponds to the viscosity. We impose periodic boundary conditions with the initial condition $u_0(x) = v_0(x) = \exp(-(14\pi)^2(x - \pi/10)^4)$. We note that when $u_0(x) = v_0(x)$ the two variables will be equal for all time, so we can effectively consider this as a single variable problem over a two-dimensional spatial domain.

For the Burgers' full order model we follow the full order model described (Wang et al., 2016). This uses finite differences to compute the spatial derivatives and uses a fixed-time step implicit method with Newton iterations for time integration. For the full order model benchmark we choose a 1024×1024 spatial grid.

C.3. Rotating Detonating Engine

The equations for the RDE setup we investigate are given as follows:

$$\begin{aligned}
\frac{\partial}{\partial t}\eta(x,t) =& -\eta(x,t)\frac{\partial}{\partial x}\eta(x,t) + v\frac{\partial^2}{\partial x^2}\eta(x,t) \\
&+ (1 - \lambda(x,t))\omega(\eta(x,t)) + \xi(\eta(x,t)), \\
\frac{\partial}{\partial t}\lambda(x,t) =& \nu\frac{\partial^2}{\partial x^2}\lambda(x,t) + (1 - \lambda(x,t))\omega(\eta(x,t)) \\
&- \beta(\eta(x,t); \mu)\lambda(x,t).
\end{aligned}$$

The function ω which models the heat release of the system is given by,

$$\omega(\eta(x,t)) = k_{\text{pre}} e^{\frac{\eta(x,t) - \eta_c}{\alpha}}.$$

The function β describes the injection term and is given by,

$$\beta(\eta(x,t); \mu) = \frac{\mu}{1 + e^{r(\eta(x,t) - \eta_p)}}.$$

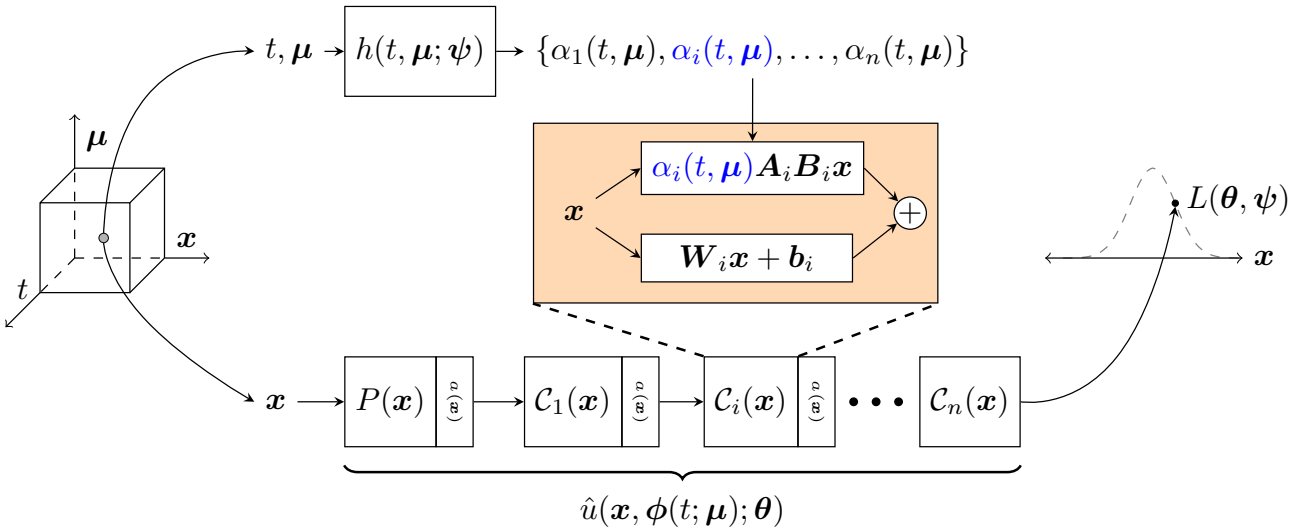

Figure 7. The CoLoRA architecture uses a hyper-network h to generate a set of continuous parameters α which are used to scale low rank matrices $\boldsymbol{A}_i \boldsymbol{B}_i$ which are internal to the reduced order model $\hat{u}$. The parameters of $\boldsymbol{\psi}$ and $\boldsymbol{\theta}$ are then jointly optimized to fit data from the full order model u_{F}.

$\xi(\eta(x,t)) = -\varepsilon\eta(x,t)$ corresponds to the energy loss of the system. We examine these equations on a circular domain $\Omega = [0, 2\pi)$ over time $\mathcal{T} = [0, 20]$. The hyperparamters for these equations are given as follows: $\nu = 0.025$, $k_{\mathrm{pre}} = 1$, $\alpha = 0.3$, $\eta_c = 1.1$, $\eta_p = 0.5$, $r = 5$, $\epsilon = 0.11$. The initial condition is given by,

$$\eta(x, 0) = 0.4 \exp(-2.25(x - \pi)^2) + 1.0$$
$$\lambda(x, 0) = 0.75\,.$$

The implementation for the RDE full order model follows (Singh et al., 2023) which uses finite differences to compute the spatial derivatives and use a fixed-time step implicit method with Newton iterations for time integration.

D. Pre-training and architecture details

As stated in Section 6.2, the reduced-model parameterization $\hat{u}$ is a multilayer perceptron with CoLoRA layers. There are 8 layers with swish nonlinear activation functions in between each layer. The first layer is a periodic embedding layer as described in Appendix D.1 which ensures the network obeys the periodic boundary condition of the PDEs we consider. This leaves the 7 subsequent layers available to be either CoLoRA layers or standard linear layers. If the dimension of the online parameters are less than 7 ($q < 7$), then the CoLoRA layers are the first q most inner layers in order to increase their nonlinear effect. For all $\mathcal{C}$ layers, the rank is $r = 3$, unless otherwise stated.

The width of all layers is 25 except the last whose width must be 1 in order to output a scalar field. The only larger network is used in the 3D spherical shallow water example where the width is 128. In the case of the RDE example and the 2D Wave example given in Yin et al. (2023) the last layer is of width 2 in order to output a field for each variable in the equation. The hyper-network h is a multilayer perceptron of depth 3 which also uses swish nonlinear activation function. The width of each layer is 15, except the last layer whose width is q, the dimension of the online parameter vector $\boldsymbol{\phi}(t, \boldsymbol{\mu})$.

D.1. Periodic P layer

All the equations we consider here have periodic boundary conditions. These can be enforced exactly by having the first layer of $\hat{u}$, which we call P, embed the $\boldsymbol{x}$ coordinates periodically. For an input $\boldsymbol{x} \in \mathbb{R}^q$ a P layer with period ω is defined as

$$P(\boldsymbol{x}) = \sum_{i=1}^{d} \left[a \cos(\boldsymbol{x} \frac{2\pi}{\omega} + c) + b \right]_i$$

Method	Vlasov (rel. err.)	2D Burgers (rel. err.)	RDE (rel. err.)
High Data (100 Trajectories)			
F-FNO	8.57e-3	5.11e-3	2.21e-3
CoLoRA-EQ	1.58e-3	2.27e-3	1.49e-3
CoLoRA-D	9.87e-4	4.96e-4	2.05e-4
Low Data (10 Trajectories)			
F-FNO	7.48e-2	2.40e-2	5.69e-3
CoLoRA-EQ	2.73e-3	3.99e-3	1.79e-3
CoLoRA-D	2.37e-3	1.76e-3	4.47e-4

Table 2. Detailed results of F-FNO data efficiency experiment

where $a, c, b \in \mathbb{R}^d$ are additionally part of the offline parameters $\boldsymbol{\theta}$. The only exception is in 1D Inviscid Burgers' given in (Chen et al., 2023b) which does not have periodic boundary conditions. Here we simply replace P with another $\mathcal{C}$ layer. In this case the boundary are loosely enforced via pretraining.

D.2. Normalization

The hyper-network given by h normalizes its input so that $\boldsymbol{\mu}$ and t are mean zero and standard deviation 1, where these statistics are computed across the training data. The reduced model given by $\hat{u}$ normalizes the $\boldsymbol{x}$ coordinates so that they are fixed between [0, 1]. The period of the periodic layer is then set to 1 in order to correspond to the normalized data.

D.3. Pre-training

In pre-training for all our benchmark problems (Vlasov, Burgers', and RDE) we minimize (6) using an Adam optimizer (Kingma & Ba, 2015) with the following hyper-parameters,

- learning rate : $5\text{e}{-3}$
- scheduler : cosine decay
- β_1: 0.9
- β_2: 0.999

For the results given in Table 1 for the 2D Wave and 3D Shallow Water problems, we use $250{,}000$ and $2{,}000{,}000$ iterations respectively, with all other hyper-parameters kept the same.

E. F-FNO experiments

For implementation of the F-FNO we use the code base given in the original paper (Tran et al., 2023) while keeping the modification that we make minimal. We use their largest architecture which is 24 layers deep as this was shown to give the best possible performance on their benchmarks. This was obtained via a grid sweep of the number of layers and time step size for the F-FNO. Additionally we give our $\boldsymbol{\mu}$ as input to their network. We train over 100 epochs as in their implementation. In order to give the F-FNO the best possible performance, the error reported is from the best possible checkpoint over all the epochs. All other hyper-parameters we set according to their implementation.

We provide additional results of the experiments in Table 2.

F. Linear projection as comparison with respect to best approximation error

In order to compute the optimal linear projection we assemble the training and test data into two snapshot matrices. We then compute the singular value decomposition of the training snapshot matrix and build a projection matrix from the top n left singular values where n is the reduced dimension. We then use this projection matrix to project the test data into the reduce space and then project back up into the full space using the transpose of the projection matrix. We then measure the relative error between the resulting project test data and the original test data. This value gives the optimal linear approximation error. For additional details see (Kramer et al., 2024).

G. Sampling train and test trajectories

Section 6.5 examines the performance of CoLoRA against an F-FNO and linear interpolation as one increases the number of training trajectories. In order to appropriately run this experiment we need a consistent way of sampling the training trajectories from the $\boldsymbol{\mu} \in \mathcal{D}$ ranges we examine. We first generate many trajectories from equidistant-spaced parameters in our range $\mathcal{D}$. This is our total trajectory dataset. We then pick three test trajectories from this set which are equally spaced out. Then as we increase the number of training trajectories (i.e. the value on the x-axis of Figure 4), we pick trajectories from our total trajectory dataset so as to maximize the minimum distance of any training trajectory from any test trajectory. This ensures that as we increase the number of training trajectories the difficulty of the problem (from an interpolation perspective) decreases.

For Burgers' we generate 101 equidistant samples of $\boldsymbol{\mu}$ in the range $\mathcal{D} = [0.01, 0.001]$. For Vlasov we generate 101 equidistant samples of $\boldsymbol{\mu}$ in the range $\mathcal{D} = [0.2, 0.4]$. The test samples for Burgers' are $[0.00253, 0.0055, 0.00847]$. The test samples for Vlasov are $[0.234, 0.3, 0.366]$.

For all other experiments the train-test splits are as follows:

Equation	**Train**	**Test**
Vlasov	$[0.2, 0.224, 0.274, 0.3, 0.326, 0.376, 0.4]$	$[0.25, 0.35]$
Burgers	$[0.001, 0.00199, 0.00298, 0.00496, 0.00595, 0.00694, 0.00892, 0.01]$	$[0.00397, 0.00793]$
RDE	$[2.0, 2.1, 2.2, 2.4, 2.5, 2.6, 2.8, 2.9, 3.0, 3.1]$	$[2.3, 2.7]$

H. Neural Galerkin computational procedure

At each time step, for samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{n_x} \in \Omega$, the computational procedure of Neural Galerkin schemes forms the batch gradient matrix $\boldsymbol{J}(\boldsymbol{\phi}(t;\boldsymbol{\mu})) \in \mathbb{R}^{n_x \times q}$ with respect to the online parameters,

$$\boldsymbol{J}(\boldsymbol{\phi}(t;\boldsymbol{\mu})) = [\nabla_{\phi_1} \hat{u}(\boldsymbol{x}_1;\boldsymbol{\theta},\boldsymbol{\phi}(t;\boldsymbol{\mu})), \ldots, \nabla_{\phi_q} \hat{u}(\boldsymbol{x}_m;\boldsymbol{\theta},\boldsymbol{\phi}(t;\boldsymbol{\mu}))]^T$$

and the n_x-dimensional vector $\boldsymbol{f}(t,\boldsymbol{\phi}(t,\boldsymbol{\mu})) = [f(t,\boldsymbol{x}_1;\hat{u}(\cdot;\boldsymbol{\theta},\boldsymbol{\phi}(t,\boldsymbol{\mu})), \ldots, f(t,\boldsymbol{x}_m;\hat{u}(\cdot;\boldsymbol{\theta},\boldsymbol{\phi}(t,\boldsymbol{\mu})))]$. The batch gradient and right-hand side lead to the linear least-squares problem in $\dot{\boldsymbol{\phi}}(t,\boldsymbol{\mu})$,

$$\min_{\dot{\boldsymbol{\phi}}(t;\boldsymbol{\mu})} \|\boldsymbol{J}(\boldsymbol{\theta},\boldsymbol{\phi}(t;\boldsymbol{\mu}))\dot{\boldsymbol{\phi}}(t;\boldsymbol{\mu}) - \boldsymbol{f}(\boldsymbol{\theta},\boldsymbol{\phi}(t;\boldsymbol{\mu}))\|_2^2\,, \tag{7}$$

which is then discretized in time and solved for the corresponding trajectory of latent states $\boldsymbol{\phi}(t_1,\boldsymbol{\mu}), \ldots, \boldsymbol{\phi}(t_K,\boldsymbol{\mu}) \in \mathbb{R}^q$ at the time steps $t_1 < \cdots < t_K$. We refer to (Bruna et al., 2024; Berman & Peherstorfer, 2023) for details on this computational approach.