Efficient Learning of PDEs via Taylor Expansion and Sparse Decomposition into Value and Fourier Domains

Md Nasim , Yexiang Xue

Department of Computer Science, Purdue University, West Lafayette, IN, USA
mnasim@purdue.edu, yexiang@purdue.edu

Abstract

Accelerating the learning of Partial Differential Equations (PDEs) from experimental data will speed up the pace of scientific discovery. Previous randomized algorithms exploit sparsity in PDE updates for acceleration. However such methods are applicable to a limited class of decomposable PDEs, which have sparse features in the value domain. We propose REEL, which accelerates the learning of PDEs via random projection and has much broader applicability. REEL exploits the sparsity by decomposing dense updates into sparse ones in both the value and frequency domains. This decomposition enables efficient learning when the source of the updates consists of gradually changing terms across large areas (sparse in the frequency domain) in addition to a few rapid updates concentrated in a small set of "interfacial" regions (sparse in the value domain). Random projection is then applied to compress the sparse signals for learning. To expand the model applicability, Taylor series expansion is used in REEL to approximate the nonlinear PDE updates with polynomials in the decomposable form. Theoretically, we derive a constant factor approximation between the projected loss function and the original one with poly-logarithmic number of projected dimensions. Experimentally, we provide empirical evidence that our proposed REEL can lead to faster learning of PDE models (70%-98% reduction in training time when the data is compressed to 1% of its original size) with comparable quality as the non-compressed models.

1 Introduction

Physics models encoded in Partial Differential Equations (PDE) are crucial in the understanding of many natural phenomena such as sound, heat, elasticity, fluid dynamics, quantum mechanics etc. Learning these physics models from data is essential to enhance our understanding of the world, accelerate scientific discovery, and design new technologies. Deep neural networks (e.g., physics-informed neural nets (Raissi, Perdikaris, and Karniadakis 2019b), Hamiltonian neural nets (Greydanus, Dzamba, and Yosinski 2019)) have been successfully deployed in this domain. In these approaches, the spatial and temporal updates of a PDE model are matched with ground truth experimental observations. A loss function is defined based on the mismatch between the simulation and the ground truth, and then the physics model is learned by back-propagation of error gradients of the loss function (Xue et al. 2021a; Raissi, Perdikaris, and Karniadakis 2019b). Nevertheless, such learning processes are expensive because of the need to back-propagate gradients over spatial and temporal simulations involving millions of mutually interacting elements.

One line of successful approaches to accelerate the learning of PDE models exploits the sparsity nature of system changes over time. For example, during the microstructure

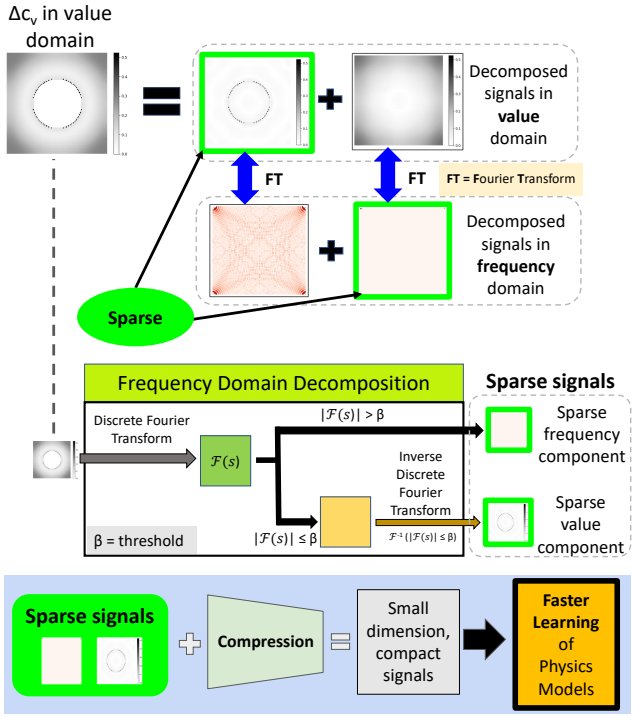


Figure 1: High-level idea of our REEL algorithm. REEL decomposes spatial and temporal updates into sparse signals in the value and the frequency domains. Random projection compresses these sparse signals for accelerated learning of PDEs. This example uses the decomposition of the vacancy concentration c_v in learning the nanovoid evolution in materials under extreme conditions.

evolution of many engineering materials, only the boundary of the microstructure changes while large portion of the system remains unchanged. It is also assumed that the corresponding PDE models can be decomposed into affine function of parameter functions and feature functions (Nasim et al. 2022; Sima and Xue 2021). The combination of decomposablity of the PDE model and sparse changes/updates over time together create opportunities for efficient algorithms which handle learning in compressed spaces using random projections and/or locality sensitive hashing. Nevertheless, such decomposablity structure applies to a limited class of PDEs and sparsity structures may change with varying initial and boundary conditions (BC/IC).

This paper propose a more general approach for efficiently learning PDE models via random projection, by exploiting sparsity in both value domain and frequency domain, and also approximating non-decomposable functions with decomposable polynomials. We observe that, systems modeled by PDEs often have slow and gradual updates across wide regions in addition to a few rapid changes concentrated in small "interfacial" regions. Such systems are frequently found in the real world. For example, during manufacturing processes such as laser sintering of powder materials into dense solids, grain boundary changes sharply at the interface area (sparse local change), while temperature rises gradually around the whole material (dense global change).

Systems with dense global change and sharp interface change limit the application of existing approaches (Nasim et al. 2022; Sima and Xue 2021) for efficiently learning relevant PDE models. However, we observe that these temporal change signals *can again become sparse if they are decomposed in value and frequency domains*. The Fourier uncertainty principle (Folland and Sitaram 1997) implies that a signal sparse in the value domain should be dense in the frequency domain and vice versa. As a result, this decomposition *can capture the sparse side of signals, whether they are in the value or the frequency domain.*

We propose Random Projection based Efficient Learning (REEL), a general approach to expedite the learning of PDEs, using signal decomposition into value and frequency domains, polynomial approximation with Taylor series, and compression via random projection. The key innovation of REEL is the inclusion of a signal decomposition step in the PDE learning framework. With this step, we convert dense value domain system updates into sparse signal components in the value and frequency domains. We also use polynomial approximation with Taylor series to approximate PDE models, which otherwise cannot be written in the decomposable form of parameter functions and feature functions. An example is the phase field model of sintering of powder compacts (Zhang and Liao 2018). After decomposition, the sparse signal components in the value and frequency domains are compressed to smaller dimensions by random projection. The learning of PDE models is then carried out in the compressed space. Notice that both the signal decomposition and compression steps are carried out once as a preprocessing step and in parallel, thus adding little computation overhead. An overview of REEL is shown in Figure 1.

Theoretically, we show the sparse projection into the value and frequency domains biases learning in a limited way. We derive a constant factor approximation bound between the projected loss function and the original one with poly-logarithmic number of projected dimensions. Experimentally, we evaluate our approaches in several real-world problems. The first is laser sintering of materials, which involves both grain boundary and temperature changes. A second application is nanovoid defect evolution in materials under irradiation and high temperature, in which both void surface movement and the emergence of interstitial and vacancy densities are considered. We demonstrate that using our REEL algorithm leads to $70 - 98\%$ reduction in training times when the data is compressed to 1% of its original size and the learned models' performance are comparable to baseline.

Our contributions can be summarized as follows : 1) we propose an efficient method to learn PDEs that have both sparse and dense feature functions, 2) we extend the applicability of random projection on sparse functions to both sparse and dense functions, using an appropriate decomposition of representation into both the value and frequency domains, 3) we extend the applicability of random projection for learning PDE models that are not readily decomposable, by using Taylor series approximation, and 4) we show empirical evidence that our learning method REEL can greatly accelerate the current state of PDE model learning.

2 Background

Partial Differential Equation (PDE) Models. PDE models of different orders appear in various scientific fields. Here, order refers to the highest derivative that appears in the PDE. Although our method is in principle applicable to PDEs of any orders, for simplicity we keep our discussion limited to the following formulation of PDEs with first order time derivative:

$$\frac{\partial u(\vec{p},t)}{\partial t} = F(u, \nabla u, \nabla^2 u, \ldots, \theta) \tag{1}$$

Here, $u(\vec{p},t)$ is a function of space $\vec{p}$ and time t, θ is a set of scalars and F is a function of both u and θ, and contains spatial derivative terms such as $\nabla, \nabla^2, \ldots$ representing first, second and even higher order spatial derivatives. u can be system state variable such as concentration, temperature etc., θ can be some system specific properties such as gradient coefficients, mobility parameter for a particular material etc. Different forms of F denotes different system models, and for a particular model, different values of θ leads to different system dynamics.

Learning PDE Models from Data. Suppose we have a sequence of ground truth PDE trajectories $u^{GT}(t)$ for time $t = 1, 2, \ldots, T$, extracted from data, and a general form of PDE model as given in Equation 1, parameterized by θ. Our goal is to learn these θ parameters.

PDE model parameters θ can be learned via numerical simulation (Xue et al. 2021a). To do so, we first replace the PDE derivatives with finite difference quotients i.e. $\frac{\partial u}{\partial t} \approx \frac{\Delta u}{dt}$ in the original equation. At any time t, we can compute the model predicted system change $\Delta u(t)$ within time interval dt, by solving Equation 1 parameterized by θ. Additionally, From data, we can extract the ground truth system

state change $\Delta u^{GT}(t) = u(t + dt) - u(t)$. Thus, we have the following at hand:

- $\Delta u^{GT}(t)$, ground truth system state change,
- $\Delta u(t)$, predicted system change from PDE model

With ground truth values of θ, both $\Delta u^{GT}(t)$ and $\Delta u(t)$ should match together. Hence we can define a loss function as follows:

$$\min_{\theta} \; L(\theta) = \sum_{t=1}^{T} ||\Delta u(t) - \Delta u^{GT}(t)||_2^2. \tag{2}$$

The loss function in Equation 2 penalizes the difference between simulation output and ground truth observation. With this neural PDE model, we can now backpropagate the error gradients from Equation 2 and update θ using stochastic gradient descent (SGD). When learning converges and the loss becomes sufficiently small, the model discovers a set of parameters θ which leads to similar dynamics as the empirical observations.

Efficient Learning of Sparse and Decomposable PDE. Sparse and Decomposable PDEs (SD-PDE) are a special class of PDEs, where temporal updates of the PDE model (i.e. $\frac{\partial u}{\partial t}$ in Equation 1) can be formulated as combination of sparse feature functions of system state variables u, multiplied by functions of learnable PDE model parameters θ. This special class of PDEs is first introduced in (Nasim et al. 2022), and the authors proposed RAPID-PDE algorithm, to accelerate the learning of corresponding PDE model parameters θ using random projection.

To learn SD-PDE models efficiently using random projection, a one-time projection of feature functions and system changes is performed. This random projection compresses the high-dimensional signals (features and system changes) into compressed low-dimensional space. Training epochs are then carried out in this compressed space. The sparsity of the high dimensional signals makes it possible to represent them with compact low dimensional signals, and the reduction of data dimension greatly accelerates the discovery of PDE model parameters.

Limitations of RAPID-PDE. Although RAPID-PDE can be highly efficient in learning PDE models, its applicability is limited by the decomposability and sparsity structure of the PDE models. Feature functions of the PDE model instances can contain dense value domain signals, depending upon initial and boundary conditions. Moreover, not all PDE models can be decomposed readily into inner products of feature functions and parameter functions.

3 REEL: Efficient Learning of PDEs with Polynomial Approximation and Signal Decomposition into Value and Frequency Domains

Our REEL algorithm aim to accelerate learning of a broad class of PDE models, which 1) may not be readily decomposable into inner product parameter function and feature functions, and/or 2) may have feature functions/system changes that are not sparse in value domain. Before moving to the details of our REEL algorithm, let us first take a close look at the formulation of SD-PDE models we mention in Section 2, which have the following general format:

$$\frac{\partial u(\vec{p}, t)}{\partial t} = \vec{\phi}(\theta)\vec{W}(u) = [\phi_1(\theta), \phi_2(\theta), \ldots, \phi_n(\theta)] \begin{bmatrix} W_1(u) \\ W_2(u) \\ \vdots \\ W_n(u) \end{bmatrix} \tag{3}$$

Here, u is the system state variable of interest, W_i are feature functions of u, often sparse in value domain, and independent of PDE model parameters θ. Similarly, ϕ_i are functions of θ and independent of u. Exact forms of ϕ_i and W_i depend on the PDE model. By replacing derivatives with finite difference quotients, and using a random matrix $\boldsymbol{P}$ for random projection on both sides, we can rewrite Equation 3 as follows:

$$\frac{\boldsymbol{P}\Delta u}{dt} = \boldsymbol{P}[\phi_1(\theta), \phi_2(\theta), \ldots, \phi_n(\theta)] \begin{bmatrix} W_1(u) \\ W_2(u) \\ \vdots \\ W_n(u) \end{bmatrix}$$

$$\Rightarrow \boldsymbol{P}\Delta u = dt.[\phi_1(\theta), \phi_2(\theta), \ldots, \phi_n(\theta)] \begin{bmatrix} \boldsymbol{P}W_1(u) \\ \boldsymbol{P}W_2(u) \\ \vdots \\ \boldsymbol{P}W_n(u) \end{bmatrix} \tag{4}$$

where, $\boldsymbol{P}\Delta u$ is the *predicted* compressed system state changes, and $[\boldsymbol{P}W_1(u), \ldots, \boldsymbol{P}W_n(u)]$ are the compressed features. The ground truth system state change Δu^{GT} is also compressed to $\boldsymbol{P}\Delta u^{GT}$ by one-time random projection with $\boldsymbol{P}$. RAPID-PDE algorithm then learns θ in the compressed space – minimizing a loss function that penalizes the difference between $\boldsymbol{P}\Delta u$ and $\boldsymbol{P}\Delta u^{GT}$.

Our REEL learning algorithm is inspired from a few observations. **First**, we notice that although it is not possible to decompose all PDE models as in Equation 3, nonlinear functions can be approximated by polynomials with Taylor series expansion and then written in decomposable form. For example, $\sin(u\theta) \approx u\theta - \frac{(u\theta)^3}{3!} + \frac{(u\theta)^5}{5!} - \frac{(u\theta)^7}{7!}$, and this approximation can be written similar to Equation 3. **Second**, for decomposed PDE models, the system change Δu and the feature functions W_i in Equation 3 may not be sparse in value domains; however, a change of representation domains, i.e. combination of value domain and frequency domain can make these signals sparse. For example, in the application domain considered in this work, the sintering of powder particles with high energy heat source such as laser, heat gets diffused into the particles and surroundings, and the particles fuse together. During this sintering process, changes in the particles happen at the boundary (sparse value domain update), while the change in specimen temperature due to heat diffusion is more widespread across the whole specimen (dense value domain update). In practice, the dense temperature change, and similarly many other dense value domain signals can be represented by sparse frequency domain signals by applying Fourier transform.

Using these two techniques of polynomial approximation and signal decomposition with Fourier transform, our REEL algorithm transforms PDE models into decomposable

PDEs with sparse value and sparse frequency domain feature functions, and then use random projection to compress the sparse signals. We then use these compressed signals to learn the PDE model parameters. We now describe our REEL learning framework in more details.

Taylor Series Approximation

For PDE models that are not decomposable into parameter functions and feature functions, we use Taylor series approximation upto certain order terms and the resulting polynomial can then be written as the decomposable form in Equation 3. Assuming that the function F in Equation 1 is infinitely differentiable at $\theta = a$, and dropping the spatial derivatives $\nabla u, \nabla^2 u, \ldots$ to avoid cumbersome notation, we can write:

$$\begin{aligned}\frac{\partial u(\vec{p},t)}{\partial t} &= F(u, \nabla u, \nabla^2 u, \ldots, \theta) \\ &\approx F(\theta = a) + (\theta - a)\frac{\partial F}{\partial \theta}\Big|_{\theta=a} \\ &+ \frac{1}{2!}(\theta - a)^2 \frac{\partial^2 F}{\partial \theta^2}\Big|_{\theta=a} + \ldots \\ &= [1, (\theta - a), \frac{1}{2}(\theta - a)^2, \ldots] \begin{bmatrix} F \\ \frac{\partial F}{\partial \theta} \\ \frac{\partial^2 F}{\partial \theta^2} \\ \vdots \end{bmatrix}_{\theta=a}\end{aligned}$$

This is the same form as in Equation 3. Approximation with Taylor expansion comes with approximation errors which we quantify with the following theorem:

Theorem 3.1. *(Proof in supplementary material) If $F(\theta)$ is at least $(n+1)$ time differentiable around $\theta = a$, except possibly at a, then $F(\theta) = \sum_{i=0}^{n} \frac{(\theta-a)^n}{n!} \frac{\partial^n F}{\partial \theta^n}\Big|_{\theta=a} + E_n$, where the approximation error $E_n = \frac{(\theta-a)^{n+1}}{(n+1)!} \frac{\partial^n F}{\partial \theta^n}\Big|_{\theta=c}$ for a suitable c in the closed range joining θ and a.*

Theorem 3.1 is a reformulation of Lagrange's remainder, and provides us error bound E_n for n-th order Taylor series approximation. In practice, we use empirical testing to decide on the order of the polynomial.

Signal Decomposition with Fourier Transform

Algorithm 1: Value and Frequency Domain Decomposition (VFDD)

1: **Input:** Signal s, threshold β
2: **Output:** Frequency domain component s_{freq} and value domain component s_{val} of signal s
3: Computer $\mathcal{F}(s)$, the Discrete Fourier Transform of s;
4: $s_{freq} \leftarrow \mathcal{F}(s) \times \mathbf{1}_{|\mathcal{F}(s)|>\beta}$;
5: $s_{val} \leftarrow \mathcal{F}^{-1}(\mathcal{F}(s) \times \mathbf{1}_{|\mathcal{F}(s)|\leq\beta})$;

With a PDE model that is now decomposed into parameter function and feature function as in Equation 3, we now look into sparsity structure of the problem. Let $\Delta u^{GT}(t)$ be the ground truth change in system variable u at time t. We use value and frequency domain signal decomposition as outlined in Algorithm 1 to convert dense $\Delta u^{GT}(t)$ signals into combination of sparse value and sparse frequency signals.

In Algorithm 1, given an input signal and a threshold, we first compute the discrete Fourier transform of the signal (line 3). We then separate the high coefficient frequency terms from the low ones based on the threshold (line 4). Using inverse discrete Fourier transform, we then convert the low coefficient frequency terms back to value domain (line 5). Note that both value and frequency domain components have the same size as the original signal, with a portion of them zeroed out based on the threshold, which is chosen by empirically testing few signal samples.

After value and frequency domain decomposition, $\Delta u^{GT}(t)$ is separated into sparse frequency component $\Delta u^{GT}_{freq}(t)$ and sparse value component $\Delta u^{GT}_{val}(t)$. On broad stroke, $\Delta u^{GT}_{freq}(t)$ corresponds to dense but slow background change of $u^{GT}(t)$, while $\Delta u^{GT}_{val}(t)$ corresponds to sharp but small interfacial change. In a similar way, we can also convert W_i, the feature function in Equation 3, into separate frequency and value domain components $W_{i(freq)}$ and $W_{i(val)}$. Then the PDE model predicted value domain system state change $\Delta u_{val}(t)$ can be computed as:

$$\Delta u_{val}(t) = dt.[\phi_1(\theta), \phi_2(\theta), \ldots, \phi_n(\theta)] \begin{bmatrix} W_{1(val)}(u) \\ W_{2(val)}(u) \\ \vdots \\ W_{n(val)}(u) \end{bmatrix}$$

The frequency domain change $\Delta u_{freq}(t)$ can be computed in similar manner. The parameter functions ϕ_i are same for both value domain and frequency domain components.

Signal Compression with Random Projection

After signal decomposition into sparse value and sparse frequency domain components, we use one-time random projection with random matrix $\boldsymbol{P}$ to compress all the sparse signal components for both system state change (yielding $\boldsymbol{P}\Delta u^{GT}_{val}, \boldsymbol{P}\Delta u^{GT}_{freq}$) and feature functions (yielding $\boldsymbol{P}W_{i(val)}, \boldsymbol{P}W_{i(freq)}$). Then the compressed feature functions can be used to compute predicted compressed system change using Equation 4. Finally, to learn PDE model parameters θ, we minimize the following loss:

$$\begin{aligned}L_{\text{REEL}}(\theta) = &\sum_{i=1}^{T} ||\boldsymbol{P}\Delta u_{val}(t) - \boldsymbol{P}\Delta u^{GT}_{val}(t)||_2^2 \\ &+ \lambda||\boldsymbol{P}\Delta u_{freq}(t) - \boldsymbol{P}u^{GT}_{freq}(t)||^2. \quad (5)\end{aligned}$$

Here, λ is a hyperparameter. The loss function in Equation 5 consists of two parts. The first part inside the summation penalizes the difference between the predicted compressed value domain change and ground truth compressed value domain change, while the second part penalizes the difference between the frequency domain counterparts. We can then use stochastic gradient descent to find the optimal parameters θ that minimize loss $L_{\text{REEL}}(\theta)$ in Equation 5.

Theorem 3.2. *(Proof in supplementary material) Suppose the projection matrix $\boldsymbol{P} = (p_{i,j})_{n\times d}$, $p_{i,j} =$*

$y_{i,j}/\sqrt{n}$. $y_{i,j}$ are sampled i.i.d. from a given distribution. $y_i^T = (y_{i,1}, \ldots, y_{i,d})$, $Y = (y_1, \ldots, y_n)^T$. $E(y_{i,j}) = 0$, $Var(y_{i,j}) = 1$. *For any x, $||y_i^T x||^2/||x||_2^2$ is sub-exponential with parameter (σ^2, b). After value and frequency domain decomposition as outlined in Algorithm 1, let $\Delta u_{val}^{GT}(t)$ and $\Delta u_{val}(t)$ have at most k_1 non-zero elements, and all $\Delta u_{freq}^{GT}(t)$ and $\Delta u_{freq}(t)$ have at most k_2 non-zero elements. $2k_1 < n, 2k_2 < n$. $0 < \delta < \min\{1, \sigma^2/b\}$. We separate the loss function in value domain and in frequency domain without random projection: $L_{val} = \sum ||\Delta u_{val}(t) - \Delta u_{val}^{GT}(t)||^2$, $L_{freq} = \sum ||\Delta u_{freq}(t) - \Delta u_{freq}^{GT}(t)||^2$. Suppose θ^* is the optimal parameter which minimizes $L_T(\theta) = L_{value} + \lambda L_{freq}$, i.e., $\theta^* = \arg\min L_T(\theta)$. Then with probability at least $[1 - 2(12/\delta)^{2k_1} \exp(-n\delta^2/(8\sigma^2))][1 - 2(12/\delta)^{2k_2} \exp(-n\delta^2/(8\sigma^2))]$, we have:*

$$(1-\delta)^2 L_T(\theta^*) \leq L_{\text{REEL}}(\theta^*) \leq (1+\delta)^2 L_T(\theta^*). \quad (6)$$

On the opposite side, suppose θ' is the local optimal solution found by REEL*, with the same probability we have:*

$$(1-\delta)^2 L_T(\theta') \leq L_{\text{REEL}}(\theta') \leq (1+\delta)^2 L_T(\theta'). \quad (7)$$

In layman terms, Theorem 3.2 implies that random projection in value and frequency domain has limited effect on learning provided that the signals are sufficiently sparse after value and frequency domain decomposition, and we only require poly-logarithmic number of projected dimensions for constant factor approximation.

4 Related Works

Learning Dynamics Models. Machine learning to learn physics dynamics models have been a popular research domain in recent years. Recently, learning Partial Differential Equations (PDEs) from data has also been studied extensively (Dzeroski and Todorovski 1995; Brunton, Proctor, and Kutz 2016; Wu and Tegmark 2019; Zhang and Lin 2018; Iten et al. 2020; Cranmer et al. 2020b; Raissi, Yazdani, and Karniadakis 2020; Raissi, Perdikaris, and Karniadakis 2019a; Liu and Tegmark 2021; Xue et al. 2021b; Chen et al. 2018). Many of the existing works in learning dynamics from data combines domain knowledge with machine learning models such as neural networks. Some of the notable works include ((Sirignano and Spiliopoulos 2018),(Raissi, Perdikaris, and Karniadakis 2019b), (Lutter, Ritter, and Peters 2018), (Demeester 2019), (Long et al. 2018)), (Xue et al. 2021a) where neural networks have been used to solve PDEs for dynamic systems. Besides these, (Han, Jentzen, and E 2018; Beck, E, and Jentzen 2019; Raissi and Karniadakis 2018; Brunton, Proctor, and Kutz 2016) are some of the other notable works in PDE solution. Neural ODEs proposed in ((Chen et al. 2018)) and their variants such as (Kidger et al. 2020; Lee and Parish 2021; Jia and Benson 2019; Chen, Amos, and Nickel 2020; Yin et al. 2021) aim to learn ordinary differential equation based dynamics models.

Learning Physics Models from Data. Physics model learning from data have been explored in (Greydanus, Dzamba, and Yosinski 2019; Cranmer et al. 2020a; Lutter, Ritter, and Peters 2018; Niu et al. 2020). In regards to efficient methods for physics learning, (Xue et al. 2021a), (Sima and Xue 2021), (Nasim et al. 2022),(Bar-Sinai et al. 2019),(Schaeffer 2017) are similar to our works in that these also aim to make the learning more computationally efficient, using locality sensitive hashing, random projection, approximate derivatives and compressed sensing. However, we introduce the frequency domain decomposition in the learning pipeline, which was not used in previous approaches to learn physics models. To make learning more efficient, previously data compression methods such as principal component analysis, low rank approximation, feature selection etc. Pruning, quantization, low rank factorization, knowledge distillation are some of the popular techniques used to compress deep neural networks previously (Choudhary et al. 2020). In our work, we focus on data compression, with the added change of representation to both value and frequency domain.

5 Experimental Results

Solid-State Selective Laser Sintering

Selective laser sintering (SLS) is a widely used important manufacturing process, where laser energy is used to sinter powder particles into dense solid structures. Accurate modeling of microstructure evolution during sintering is very important for process control and optimization. For our experiment, we work with the phase field model of solid-state sintering as proposed in (Zhang and Liao 2018), where a thermal model is coupled with microstructure model.

Thermal Model. Heat diffusion from laser energy source during sintering process can be modeled with transient heat conduction equation as follows:

$$\rho \frac{\partial C_p T}{\partial T} = \nabla.(k \nabla T) + Q. \quad (8)$$

Here, T represents the temperature field in the specimen. ρ, C_p, k are the density, specific heat and thermal conductivity of the material respectively and assumed to be material specific constants in our experiments. Q is heat flux from the laser heat source.

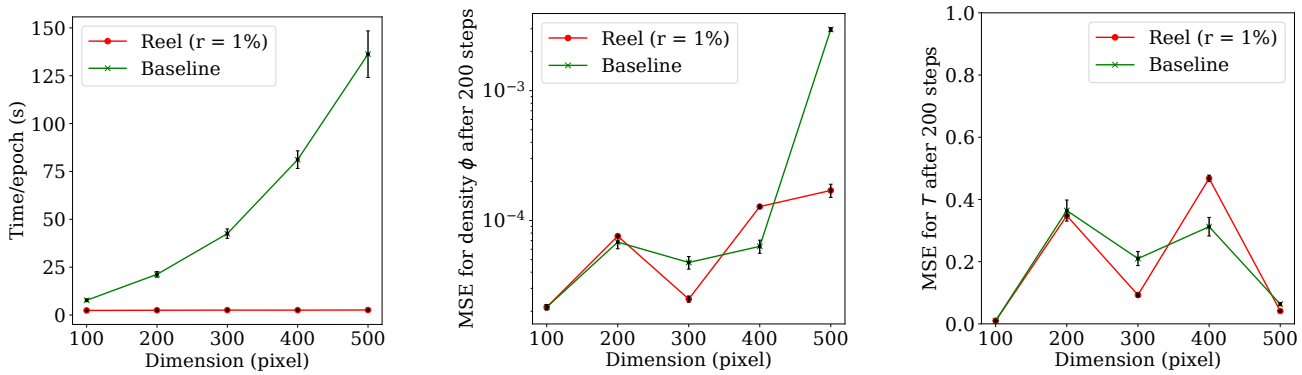
Microstructure Model. In the phase field model of sintering, the microstructure is represented with two types of field variables – conserved density field ϕ and non-conserved order parameters η_i for each particle. ϕ takes value in the range $[0, 1]$, where 1 represent solid phase and 0 represent pores. η_i similarly has value in range $[0, 1]$, and takes the value of 1 for a designated particle and 0 elsewhere. The driving force for sintering is the minimization of the total free energy F,

$$F = \int_V \left[f(\phi, \eta_{i=1,\ldots,n}) + \frac{\varepsilon_\phi}{2} |\nabla \phi|^2 + \sum_{i=1}^{n} \frac{\varepsilon_\eta}{2} |\nabla \eta_i|^2 \right] \mathrm{d}V.$$

Here, n is the number of particles in the system. The evolution of ϕ and η_i over time are governed by Cahn-Hilliard equation and Allen-Cahn equation respectively:

$$\frac{\partial \phi}{\partial t} = \nabla \cdot (M \nabla \frac{\delta F}{\delta \phi}), \qquad \frac{\partial \eta}{\partial t} = -L \frac{\delta F}{\delta \eta}, \quad (9)$$

where M and L represent atom diffusion mobility and grain boundary (GB) mobility respectively. For details of both



(a) Training times for sintering (b) MSE of ϕ for sintering (c) MSE of T for sintering

Figure 2: Our REEL algorithm saves 70% to 98% of training times for learning PDE physics models, while preserving very high accuracy of the learned models comparable to baseline (we used REEL algorithm without compression as our baseline for sintering). (a) Training times for sintering PDE model. Here, r denotes the compression level used in REEL, $r = 1\%$ means the data was compressed to 1% of the original dimension. (b-c) Mean squared error (MSE) for density (ϕ) and temperature (T) field variable in sintering model. MSE for a model was computed by performing simulation with the model for 200 timesteps, and then comparing the simulation output and ground truth. MSE of ϕ and T are very small and comparable between our REEL and baseline, and the simulation results are practically indistinguishable as shown in Figure 3.

thermal and microstructure model, we refer to the original text (Zhang and Liao 2018).

Learning Objective. Given the system states ϕ, η and T, we aim to learn the parameters of the selective laser sintering model, which are $\{C_p, \rho, k\}$ and parameters associated with F, M and L in Equation 9.

Training and Testing. For our experiment with sintering application, we used synthetic dataset according to the model in (Zhang and Liao 2018). We simulated microstructure evolution and heat diffusion during sintering in 2D for $N \times N$ grids, for $N = 100, 200, 300, 400, 500$ for $T = 20000$ timesteps. For training, we used data for 1000 timesteps. **As the baseline method, we used our REEL algorithm without the random projection step.** Stochastic gradient descent was used for optimization during training, and the learning rate was set by hyperparameter tuning. More details of training and testing (i.e. computing resources,code etc.) are provided in supplementary materials.

Nanovoid Evolution in Materials under Irradiation and High Temperature

Materials under heavy irradiation and high temperature forms many types of defects. One of these defects is named *nanovoid*, which are nano-meter scale defect clusters, formed by the accumulation of vacancy defect in crystal lattice. Such void defects greatly affects material degradation over time. Modeling the evolution dynamics of such defects are very important in designing sustainable materials that can withstand extreme environments such as inside a nuclear reactor.

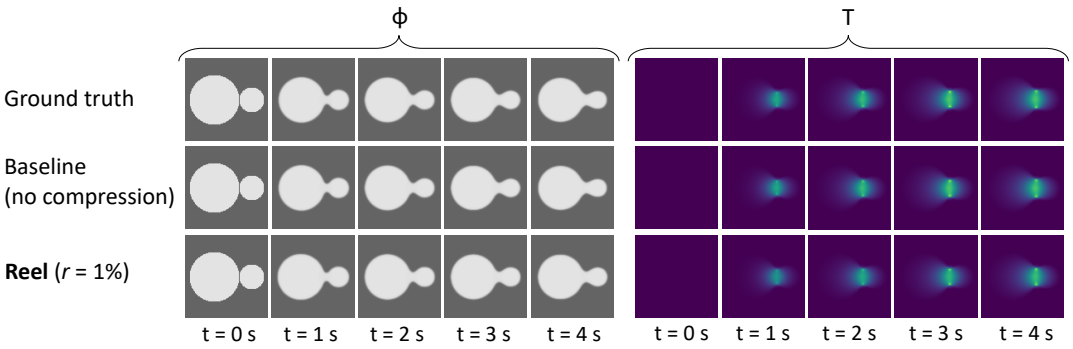
In the phase field model for nanovoid defect evolution in engineering materials, the state of a system is described by 3 phase field variables – c_v, c_i and η at each point in space and time. c_v and c_i represents the percentage of vacancy defects and interstitial defects respectively, in the crystal lattice, while η is an order parameter distinguishing between different phases of the material. For our experiment, we use the phase field PDE model proposed in (Millett et al. 2011) which describe the time evolution of phase field variables with with Cahn-Hilliard equation and Allen-Cahn equation as shown in Equation 9. For details of the phase field model, we refer to original text (Millett et al. 2011).

Training and Testing. We used a synthetic dataset generated according to the phase field model in (Millett et al. 2011) for our nanovoid experiments. For baseline, we used RAPID-PDE (Nasim et al. 2022) with two different compression levels – 1) no compression and 2) compression to 10% of original data dimension. **We used RAPID-PDE as baseline for nanovoid only, since the nanovoid PDEs are decomposable, while the sintering model PDEs are not.**

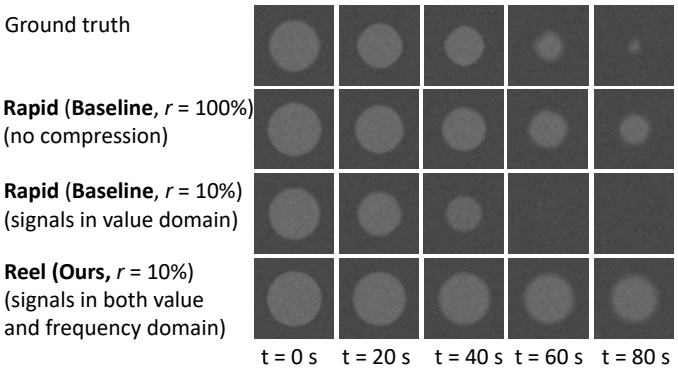
Results

Faster Learning of PDE Physics Models. Our REEL algorithm leads to faster training of PDE physics models compared to the baseline learning methods. As shown in Figure 2a, our REEL algorithm can reduce training time for sintering PDE model by $70 - 98\%$. For nanovoid PDE model training, we see a similar level of computational saving ($\approx 70\%$ reduction of training time) when no compression is used. Added to the benefit of reduced data dimension from compression, the data size also becomes small enough to perform the training epochs in memory, while for baseline method with no compression, the training epochs has to be performed with data stored in disks. We tried to train the baseline model by keeping the entire training data in memory, however the training data was too large to fit in memory. More details on experiment setups and infrastructure are provided in supplementary material.

Highly Accurate Learned Physics Models. Our REEL training algorithm leads to highly accurate PDE



(a) Simulation of particle density (ϕ) and temperature (T) change during laser sintering of powder particles



(b) Simulation of nanovoid evolution under extreme conditions

Figure 3: REEL can learn very accurate physics models governing sintering of powder particles and nanovoid defect evolution. (a) Simulation of mircrostructure evolution and heat diffusion during solid-state selective laser sintering (b) Simulation of nanovoid defect evolution in materials under extreme heat and irradiation. Here, r represents the level of data compression. For example, $r = 1\%$ means data has been compressed to 1% of original size. Our REEL algorithm can capture the slow shrinking dynamics of nanovoids as seen in the simulation (4-th row in Figure 3b), while with baseline RAPID-PDE, the void disintegrates (3-rd row in Figure 3b). Our REEL method is applicable to both PDE models, while RAPID-PDE algorithm, which is used as baseline for nanovoid experiments, cannot be applied to PDEs of sintering phase field model.

physics models. We used simulation to qualitatively evaluate the models trained with our algorithm, and the simulation results are presented in Figure 3. Overall, we find that the models trained with our REEL algorithm produce very similar dynamics as ground truth. For sintering, the baseline method with no compression also learn very accurate dynamics, however requires very high model training time as shown in Figure 2a. For quantitative evaluation, we simulated all our trained models for 200 steps using the same initial condition. This was repeated 200 times with different initial conditions, which were unseen during training. Afterwards, we computed the mean squared error (MSE) between simulation results and ground truth as shown in Figure 2b and Figure 2c. We can see that even with data dimension reduced to 1% of the original size, our REEL trained models show comparable very small error similar to baseline models. For nanovoid model, we used RAPID-PDE with different compression levels as our baseline. With data compressed to 10% of original size, REEL algorithm can capture the shrinking dynamics of void evolution as shown in Figure 3b (4-th row), while baseline RAPID-PDE algorithm results in the disintegration of void as seen in Figure 3b (3-rd row). More details of evaluation result with nanovoid PDE model is provided in supplementary material.

6 Conclusion

In this paper, we present REEL, an efficient algorithm to learn partial differential equations from experimental data. Our acceleration is based on decomposing the spatial and temporal updates into sparse signals in the value and frequency domains. REEL also uses Taylor series expansion to approximate PDE models in a decomposable form. Random projection is then applied to compress the sparse signals in the value and frequency domains as a preprocessing step. Learning is carried out entirely in the compressed space. Our method is applicable to a wide range of PDE models where the spatial and temporal updates are made of slow and wide-range changes in addition to rapid changes in the interfacial regions. Empirically, we show that our algorithm can lead to faster learning of physics models, and the learned models exhibit reasonably high accuracy on testing.

Acknowledgements

This research is primarily supported by NSF grant CCF-1918327. We also acknowledge financial support by DOE – Fusion Energy Science, under grant number: DE-SC0024583.

References

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

Beck, C.; E, W.; and Jentzen, A. 2019. Machine Learning Approximation Algorithms for High-Dimensional Fully Nonlinear Partial Differential Equations and Second-order Backward Stochastic Differential Equations. *Journal of Nonlinear Science*, 29: 1563–1619.

Brunton, S. L.; Proctor, J. L.; and Kutz, J. N. 2016. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15): 3932–3937.

Chen, R. T.; Amos, B.; and Nickel, M. 2020. Learning neural event functions for ordinary differential equations. *arXiv preprint arXiv:2011.03902*.

Chen, R. T.; Rubanova, Y.; Bettencourt, J.; and Duvenaud, D. K. 2018. Neural ordinary differential equations. *Advances in neural information processing systems*, 31.

Choudhary, T.; Mishra, V.; Goswami, A.; and Sarangapani, J. 2020. A comprehensive survey on model compression and acceleration. *Artificial Intelligence Review*, 53(7): 5113–5155.

Cranmer, M.; Greydanus, S.; Hoyer, S.; Battaglia, P.; Spergel, D.; and Ho, S. 2020a. Lagrangian neural networks. *arXiv preprint arXiv:2003.04630*.

Cranmer, M. D.; Sanchez-Gonzalez, A.; Battaglia, P. W.; Xu, R.; Cranmer, K.; Spergel, D. N.; and Ho, S. 2020b. Discovering Symbolic Models from Deep Learning with Inductive Biases. In *NeurIPS*.

Demeester, T. 2019. System Identification with Time-Aware Neural Sequence Models. *arXiv preprint arXiv:1911.09431*.

Dzeroski, S.; and Todorovski, L. 1995. Discovering Dynamics: From Inductive Logic Programming to Machine Discovery. *J. Intell. Inf. Syst.*, 4(1): 89–108.

Folland, G. B.; and Sitaram, A. 1997. The uncertainty principle: a mathematical survey. *Journal of Fourier analysis and applications*, 3: 207–238.

Greydanus, S.; Dzamba, M.; and Yosinski, J. 2019. Hamiltonian neural networks. *Advances in Neural Information Processing Systems*, 32: 15379–15389.

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

Iten, R.; Metger, T.; Wilming, H.; Del Rio, L.; and Renner, R. 2020. Discovering physical concepts with neural networks. *Physical review letters*, 124(1): 010508.

Jia, J.; and Benson, A. R. 2019. Neural jump stochastic differential equations. *Advances in Neural Information Processing Systems*, 32.

Kidger, P.; Morrill, J.; Foster, J.; and Lyons, T. 2020. Neural controlled differential equations for irregular time series. *Advances in Neural Information Processing Systems*, 33: 6696–6707.

Lee, K.; and Parish, E. J. 2021. Parameterized neural ordinary differential equations: Applications to computational physics problems. *Proceedings of the Royal Society A*, 477(2253): 20210162.

Liu, Z.; and Tegmark, M. 2021. Machine Learning Conservation Laws from Trajectories. *Phys. Rev. Lett.*, 126: 180604.

Long, Z.; Lu, Y.; Ma, X.; and Dong, B. 2018. Pde-net: Learning pdes from data. In *International Conference on Machine Learning*, 3208–3216.

Lutter, M.; Ritter, C.; and Peters, J. 2018. Deep Lagrangian Networks: Using Physics as Model Prior for Deep Learning. In *International Conference on Learning Representations*.

Millett, P. C.; El-Azab, A.; Rokkam, S.; Tonks, M.; and Wolf, D. 2011. Phase-field simulation of irradiated metals: Part I: Void kinetics. *Computational materials science*, 50(3): 949–959.

Nasim, M.; Zhang, X.; El-Azab, A.; and Xue, Y. 2022. Efficient Learning of Sparse and Decomposable PDEs using Random Projection. In *Uncertainty in Artificial Integlligence*.

Niu, T.; Nasim, M.; Annadanam, R.; Fan, C.; Li, J.; Shang, Z.; Xue, Y.; El-Azab, A.; Wang, H.; and Zhang, X. 2020. Recent Studies on Void Shrinkage in Metallic Materials Subjected to In Situ Heavy Ion Irradiations. *JOM*, 72.

Raissi, M.; and Karniadakis, G. E. 2018. Hidden physics models: Machine learning of nonlinear partial differential equations. *Journal of Computational Physics*, 357: 125–141.

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

Raissi, M.; Perdikaris, P.; and Karniadakis, G. E. 2019b. 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.

Raissi, M.; Yazdani, A.; and Karniadakis, G. E. 2020. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367(6481): 1026–1030.

Schaeffer, H. 2017. Learning partial differential equations via data discovery and sparse optimization. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2197): 20160446.

Sima, C.; and Xue, Y. 2021. LSH-SMILE: Locality Sensitive Hashing Accelerated Simulation and Learning. *Advances in Neural Information Processing Systems*, 34.

Sirignano, J.; and Spiliopoulos, K. 2018. DGM: A deep learning algorithm for solving partial differential equations. *Journal of computational physics*, 375: 1339–1364.

Wu, T.; and Tegmark, M. 2019. Toward an artificial intelligence physicist for unsupervised learning. *Phys. Rev. E*, 100: 033311.

Xue, Y.; Nasim, M.; Zhang, M.; Fan, C.; Zhang, X.; and El-Azab, A. 2021a. Physics Knowledge Discovery via Neural Differential Equation Embedding. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 118–134. Springer.

Xue, Y.; Nasim, M.; Zhang, M.; Fan, C.; Zhang, X.; and El-Azab, A. 2021b. Physics Knowledge Discovery via Neural Differential Equation Embedding. In *ECML/PKDD (5)*, volume 12979 of *Lecture Notes in Computer Science*, 118–134. Springer.

Yin, Y.; Le Guen, V.; Dona, J.; de Bézenac, E.; Ayed, I.; Thome, N.; and Gallinari, P. 2021. Augmenting physical models with deep networks for complex dynamics forecasting. *Journal of Statistical Mechanics: Theory and Experiment*, 2021(12): 124012.

Zhang, S.; and Lin, G. 2018. Robust data-driven discovery of governing physical laws with error bars. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2217): 20180305.

Zhang, X.; and Liao, Y. 2018. A phase-field model for solid-state selective laser sintering of metallic materials. *Powder Technology*, 339: 677–685.