Physics Knowledge Discovery via Neural Differential Equation Embedding

Yexiang Xue[1]✉, Md Nasim[1], Maosen Zhang[1], Cuncai Fan[2], Xinghang Zhang[3], and Anter El-Azab[3]

[1] Department of Computer Science, Purdue University, USA.
{yexiang, mnasim, maosen}@purdue.edu
[2] The University of Hong Kong, Hong Kong SAR, China.
cuncai@hku.hk
[3] School of Materials Engineering, Purdue University, USA.
{xzhang98, aelazab}@purdue.edu

Abstract. Despite much interest, physics knowledge discovery from experiment data still remains largely a manual trial-and-error process. This paper proposes neural differential equation embedding (`NeuraDiff`), an end-to-end approach to learn a physics model characterized by a set of partial differential equations directly from experiment data. The key idea is the integration of two neural networks – one recognition net extracting the values of physics model variables from experimental data, and the other neural differential equation net simulating the temporal evolution of the physics model. Learning is completed by matching the outcomes of the two neural networks. We apply `NeuraDiff` to the real-world application of tracking and learning the physics model of nano-scale crystalline defects in materials under irradiation and high temperature. Experimental results demonstrate that `NeuraDiff` produces highly accurate tracking results while capturing the correct dynamics of nano-scale defects.

Keywords: Physics Knowledge Discovery · Neural Differential Equation Embedding · Nano-scale Materials Science.

1 Introduction

The advancement and application of machine learning in the last decade has been crucial in many domains. In spite of its wide outreach, the potential to leverage machine learning for scientific discovery in a closed loop has not been fully realized. Real-world experimentation and physics-based simulation provide a *forward* approach to *validate* a given physics model. The accuracy of a hypothetical model can be verified by testing if the simulated results match actual experiments. Nonetheless, the more important *backward* learning task, namely, knowledge discovery and refinement of physics models from experimental data, remains largely a manual trial-and-error process relying on the intuitions and inspirations from the physicists (upper panel Figure 1). Recently, a series of research [16, 29, 40, 9, 39, 36] aim at learning partial differential equations from

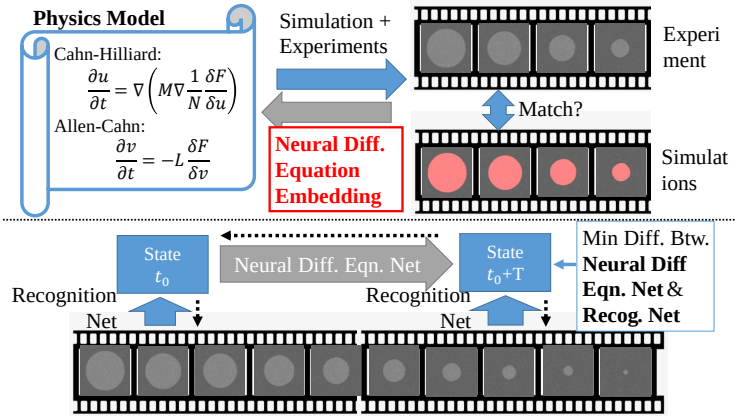


Fig. 1. (Upper) Physics experiments and simulation provides a forward approach to validate a physics model. Our Neural Differential Equation Embedding (`NeuraDiff`) is a backward approach to learn a physics model directly from experimental data. **(Lower)** The high-level idea of `NeuraDiff`. A recognition network extracts model parameters at time t_0, which are fed to a neural differential equation net to simulate evolution for T steps, and are compared with the recognized results at $t_0 + T$. Back-propagation is utilized to match the output of the recognition and the neural differential equation net.

data. However, they did not achieve fully automatic physics model identification from experiment data because the input of these models are the trajectories of differential equations, which may be unavailable from experiment data and need to be extracted as a separate step. Unfortunately this is the case in the application domain considered in this paper.

We develop neural differential equation embedding (`NeuraDiff`), an end-to-end approach to learn a physics model characterized by a set of partial differential equations directly from experiment data. The key idea is the integration of two neural networks, one neural differential equation net simulating the temporal dynamics, and the second recognition net extracting the values of physics model variables from experimental data. The high level idea is shown in the lower panel of Figure 1. Here, the recognition net extracts physics model variables at time t_0 and feed it to the differential equation net to simulate the temporal evolution for T steps. Then, the predicted model parameters are compared with the recognized values at time $t_0 + T$ and with additional annotations. Back-propagation is utilized to minimize the difference among the predictions of the recognition net, the differential equation net, and the annotations. The three predictions converge when the training is complete.

The development of `NeuraDiff` was motivated by the real-world application of tracking and learning the physics model of nano-scale crystalline defects in materials. These materials and alloys are critical for current nuclear fission reactors and future fusion devices. Nano-scale crystalline defects can appear in different forms in these materials. Extreme environments of heat and irradiation

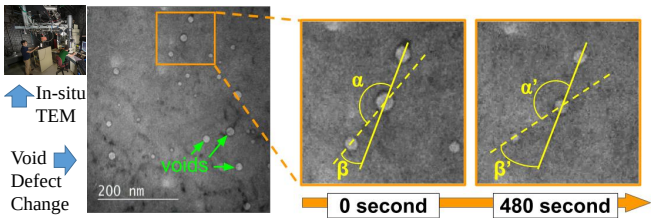


Fig. 2. (Upper Left) The Intermediate Voltage Electron Microscopy (IVEM) – Tandem Facility at the Argonne National Laboratory which provides in-situ TEM data. Source: anl.gov. **(Middle and Right)** Sample images captured during in-situ radiation experiments. The middle image shows void defects embedded in a Cu specimen at 350^oC and irradiation dose 0.25 – 1.00 dpa (dose increases with time). Void migration is illustrated by the change in sizes and in the angles of yellow lines in the right images.

can cause these defects to evolve in size and position. As shown in Figure 2, void shaped defects are captured by transmission electron microscope (TEM) cameras during in-situ radiation experiments. These defects appear in round shapes, and drift in position as demonstrated by the change of angles α, β to α', β' respectively, as time progresses. They also change size. These changes can affect the physical and mechanical properties of the material in undesirable ways as discussed in [31]. For this reason, characterizing these defects is essential in designing new materials that can resist adverse environments.

In-situ radiation experiments are carried out to analyze the evolution of crystalline defects in materials. During these experiments, changes in a material specimen, subjected to high temperature and irradiation, is recorded through a TEM camera and stored in high-resolution high frame rate videos. The huge amount of data calls for a data-driven approach to expedite the video analysis, which can bring in new scientific knowledge and insights for alloy designs. However, manual video analysis requires huge effort. According to our calculation, it takes a graduate student 3.75 months to fully annotate the defects in a 10-minute in-situ video if he spends 5 minutes per frame and devotes 40 hours a week. Phase-field modeling is a simulation tool commonly used to study the evolution of point defects. In this model, the evolution of the void shaped defects is characterized by a number of field variables. These field variables are continuous, vary rapidly at the interface of the void defects, and are governed by a number of differential equations. Data assimilation is often used to estimate the model parameters of a phase-field model from data. However, tuning phase-field models relies heavily on expert knowledge, and the results are often qualitative.

Our proposed `NeuraDiff` learns the phase field model automatically as described in [30] that governs the void nucleation and growth in irradiated materials, while provides accurate tracking of void clusters. `NeuraDiff` connects phase-field simulation and physics experiments, enabling an automatic pipeline to discover correct physics models from data. Our experimental results show that `NeuraDiff` produces highly accurate tracking results while learning the correct physics. Our model's accuracy is close to 100% on both the synthetic dataset and a

real-world in-situ dataset of Cu under 350^oC and an irradiation dose of 0.25 – 1.00 dpa (dose increases as time goes by). Moreover, our model learns the correct physics. The simulation based on the phase-field parameters learned by our model demonstrate similar dynamics as the ground truth, while a neural model without embedding physics cannot discover the correct dynamics. We also tested our model for transfer learning. Our `NeuraDiff` model correctly predicts the evolution of nano-structures from an unseen start condition while competing approaches cannot.

In summary, our contribution is as follows: 1) we propose `NeuraDiff`, an end-to-end approach integrating the recognition and the neural differential equation net to learn a physics model characterized by a set of partial differential equations directly from experiment data. 2) We apply `NeuraDiff` in tracking and learning the physics model of nano-scale crystalline defects in materials from in-situ experiments. Our approach enables detailed analysis of nano-structures at scale, which otherwise is beyond reach of manual efforts by materials scientists. 3) Our experimental results show that `NeuraDiff` produces close to 100% accuracy in tracking void defects. 4) Our `NeuraDiff` learns the correct physics while neural networks without embedding physics cannot. 5) Our `NeuraDiff` performs well in a transfer learning setting.

2 Phase-field Model

Micro-structures in nano-scale physics are spatial arrangements of the phases that have different compositional and/or structural characters; e.g., the regions composed of different crystal structures and/or having different chemical compositions, grains of different orientations, domains of different structural variants, and domains of different electric or magnetic polarizations. The size, shape, volume fraction, and spatial arrangement of these micro-structural features determine the overall properties of multi-phase and/or multi-component materials.

In a phase-field model, micro-structures are defined by a set of field variables. Field variables are assumed to be continuous and changing rapidly across the interfacial regions. For example, in the phase field model of irradiated metals, 3 different phase-filed variables c_v, c_i and η together represent the system state. $c_v(\mathbf{r}, t)$ represents the voids concentration, $c_i(\mathbf{r}, t)$ represent interstitial concentration and $\eta(\mathbf{r}, t)$ differentiates between the two phases - solid phase and void phase (details discussed later). Here, $\mathbf{r} = (x, y)$ represents the spatial coordinates and t represents time. We work with 2-dimensional case in this paper, but high dimensional cases can be handled similarly.

c_v and c_i represents two types of defects in irradiated metals – voids and interstitials. Voids result from the missing of atoms in certain crystal lattice locations, as shown in Figure 2. c_v is zero in the region consisting of 0% of voids and is one in regions of 100% voids. c_v changes continuously albeit rapidly at the interfaces of void and non-void regions. The interstitials, represented by c_i, are another variety of crystallographic defects, where atoms assume a normally unoccupied site in the crystal structure. c_i is defined similarly to c_v. The void

cluster variable η is an order parameter that spatially differentiates the 2 phases. η takes a constant value $\eta = 0$ in the solid phase and $\eta = 1$ in the void phase.

Phase-field modeling leverages a set of differential equations of these field variables to model the microstructure evolution. The temporal evolution of a conserved field variable $u(\mathbf{r}, t)$ is governed by the Cahn–Hilliard [7] equation:

$$\frac{\partial u}{\partial t} = \nabla \cdot \left(M \nabla \frac{1}{N} \frac{\delta F}{\delta u} \right). \tag{1}$$

Here, F is the free energy. M is the diffusivities of the material species and N is the number of lattice sites per unit volume of the material. $\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y} \right)$ is the diffusion operator. $\nabla \cdot \nabla$ is the laplacian i.e., $\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$. $\frac{\delta F}{\delta u}$ is the functional derivative. A non-conserved field variable v evolves according to the Allen–Cahn [1] equation:

$$\frac{\partial v}{\partial t} = -L \, \frac{\delta F}{\delta v}. \tag{2}$$

Here L is the mobility constant. In the phase-field model for irradiated metals, c_v, c_i are conserved field variables and η is a non-conserved field variable. Allen-Cahn and Cahn-Hilliard equations are the cornerstones of phase-field modeling. They offer good descriptions of the basic physics of many multi-phase systems.

Finite Difference Approach. Finite difference is a useful tool to obtain numerical solutions to differential equations. Let $(x_1, \ldots, x_{N_x})$ and $(y_1, \ldots, y_{N_y})$ be a finite discretization of the x-axis and the y-axis covering the region of interest. We use uniform step sizes, i.e., $x_i - x_{i-1} = y_j - y_{j-1} = \mathrm{d}s$ for all $i \in \{2, \ldots, N_x\}$ and $j \in \{2, \ldots, N_y\}$. As a result, the region is covered by a finite mesh of the size $N_x \times N_y$. We also assume the time is discretized into $(t_1, \ldots, t_{N_t})$ and $t_k - t_{k-1} = \mathrm{d}t$ for $k \in \{2, \ldots, N_t\}$. Let $u(\mathbf{r}, t)$ be a function that depends on location $\mathbf{r} = (x, y)$ and time t. We discretize u onto this mesh by denoting $u_{i,j,k}$ as a shorthand for $u(x_i, y_j, t_k)$. The finite difference algorithm uses the finite difference to approximate derivatives. For example, the value of $\frac{\partial u}{\partial x}(x_i, y_j, t_k)$ can be approximated by:

$$(u(x_{i+1}, y_j, t_k) - u(x_i, y_j, t_k))/(x_{i+1} - x_i) = (u_{i+1,j,k} - u_{i,j,k})/\mathrm{d}s.$$

Similarly, $\nabla^2 f$, the second order laplacian ∇^2 of a 2D function f, can be approximated by five point stencil centered second-order difference:

$$\nabla^2 f_{i,j,k} = \frac{1}{\mathrm{d}s^2} \left(f_{i+1,j,k} + f_{i-1,j,k} + f_{i,j+1,k} + f_{i,j-1,k} - 4 f_{i,j,k} \right)$$

Using this idea, both the Cahn-Hilliard and the Allen-Cahn equations can be discretized. A finite approximate solution can be obtained by simulating the evolvement of field variables from a given starting state.

3 Problem Statement

Our phase field model of irradiated metals follow largely from the work of [30]. This model incorporates a coupled set of Cahn–Hilliard and Allen–Cahn

equations to capture the processes of point defect generation and recombination, annihilation of defects at sinks. The phase-field model includes 3 field variables, c_v, c_i, and η, which vary both spatially and temporally. All of the variables are continuous, yet vary rapidly across interfaces. The total free energy F of the heterogeneous material is expressed in terms of the free energy of each constituent phases and interfaces:

$$F = N \int_V \left[h(\eta) f^s(c_v, c_i) + j(\eta) f^v(c_v, c_i) + \frac{\kappa_v}{2} |\nabla c_v| + \frac{\kappa_i}{2} |\nabla c_i| + \frac{\kappa_\eta}{2} |\nabla \eta| \right] \mathrm{d}V.$$

Here, $f^s(c_v, c_i)$ is the contribution term from the solid phase. $h(\eta) = (\eta - 1)^2$ makes sure that f^s contributes 0 when $\eta = 1$. Similarly, $f^v(c_v, c_i)$ is the contribution term from the void phase, and $j(\eta) = \eta^2$. We use the formulation from [30] for f^s and f^v:

$$f^s(c_v, c_i) = E_v^f c_v + E_i^f c_i + k_B T [c_v \ln c_v + c_i \ln c_i + (1 - c_v - c_i) \ln(1 - c_v - c_i)],$$
$$f^v(c_v, c_i) = (c_v - 1)^2 + c_i^2.$$

According to the phase-field model, the dynamics of the field variables c_v, c_i and η should follow the Cahn-Hilliard and the Allen-Cahn equations. Nevertheless, new voids and interstitials can form due to irradiation and thermal fluctuation. Therefore, the standard equations need to be updated to the form:

$$\begin{aligned}
\frac{\partial c_v}{\partial t} &= \nabla \cdot (M_v \nabla \frac{1}{N} \frac{\delta F}{\delta c_v}) + \xi(\mathbf{r}, t) + P_v(\mathbf{r}, t) - R_{iv}(\mathbf{r}, t), \\
\frac{\partial c_i}{\partial t} &= \nabla \cdot (M_i \nabla \frac{1}{N} \frac{\delta F}{\delta c_i}) + \zeta(\mathbf{r}, t) + P_i(\mathbf{r}, t) - R_{iv}(\mathbf{r}, t), \\
\frac{\partial \eta}{\partial t} &= - L \frac{\delta F}{\delta \eta} + \iota(\mathbf{r}, t) + P_v(\mathbf{r}, t).
\end{aligned}$$

Here, ξ, ζ and ι are thermal fluctuation terms, modeling the fact that voids and interstitials can appear randomly in the environments of high temperature and irradiation. P_v and P_i reflect the voids (and interstitials) introduced during the irradiation process. Irradiation hits the surface of the materials and both voids and interstitials can form as a result. R_{iv} models the cancellation of voids and interstitials. We refer the details of these terms to the original publication [30]. In this model, the following set of parameters $P = \{E_v^f, E_i^f, k_B T, \kappa_v, \kappa_i, \kappa_\eta, M_v, M_i, L\}$ determine the evolution of nanovoids. Our physics learning task is to identify the values of these parameters from a partially annotated video of void dynamics.

We assume access to partial video annotations, in which part of regions in a subset of frames are annotated. For simplicity, we assume one pixel in one frame V can be in three states: 0 means the pixel is annotated to be in a solid state; i.e., $\eta = 0$; 1 means the pixel is annotated to be part of a void cluster; i.e., $\eta = 1$; $*$ means the pixel is not annotated or the annotator is not sure of its state. We denote A as a matrix of these annotations, each entry of which is one of the three states for the corresponding pixel. The **physics-aware micro-structure tracking problem** is defined as:

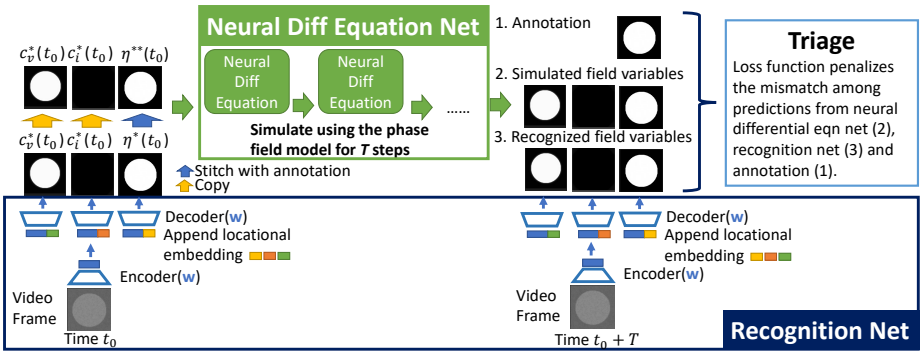


Fig. 3. The architecture of `NeuraDiff`. Our architecture consists of a recognition net, which predicts field variable values (c_v, c_i and η) based on video frames. A second neural differential equation net simulates phase field evolution for T steps. Finally, a loss function is applied which penalizes the difference among the predictions from the neural differential equation net, the recognition net and the annotations. Backpropagation is then used to train the two neural networks to minimize the loss function.

- **Given**: $\{(t_1, V_1, A_1), (t_2, V_2, A_2), \ldots, (t_N, V_N, A_N)\}$ as a partially annotated video of nano-structural evolution, where $t_1, \ldots, t_N$ are time stamps, V_i is the video frame for the time stamp t_i and A_i is the partial annotation for V_i, in which each pixel is annotated to one of the three states.
- **Find**: (i) ***track microstructures***: for each frame V_i, find matrix η_i, which contains the predicted η value for each pixel. (ii) ***learn physics***: find the set of phase-field parameters P, along with the values of the unobserved variables c_v and c_i, which best fit the micro-structure evolution.

4 Neural Differential Equation Embedding

Our `NeuraDiff` model learns a physics model directly from experiment data via a tight integration of a neural differential euqation net and a recognition net, embedding phase-field simulation into neural network learning. The high level idea is shown in Figure 3. A recognition net extracts the values of the three field variables, c_v, c_i and η from noisy video frames. Taking these field variables as initial condition, the neural differential equation net uses the finite difference method to simulate a phase-field model. We implement the finite difference method as a convolutional neural net (details discussed later), the parameters of which can be updated via back-propagation. This architecture is related to the recurrent neural networks (RNN), where the same operational step is repeatedly applied during the forward pass. Contrary to RNNs, each step in our neural differential equation net represents a simulation step of the phase-field model.

`NeuraDiff` works through a triage process. First the recognition net extracts the three field variables from the video frame at time stamp t_0. The predicted field values are partially replaced by the groundtruth annotations (if they are present at t_0) and are sent to the neural phase-field net. The neural differential equation net then simulates the phase-field model for T steps and outputs the

simulated field variable values at time $t_0 + T$. We also have partial annotations at the time $t_0 + T$ and the predictions of these field variable values from the recognition net. Ideally, if the recognition net is trained to predict the three field variables accurately and the neural differential equation net has the ground-truth parameters, then the three outcomes, namely, the simulated, the recognized field variables, and the partial annotations at the time $t_0 + T$ should match. Therefore, we enforce a loss function which penalizes the differences among the three outcomes. Back-propagation is then used to minimize this loss function. At the end of training, when the predictions from both neural nets and the partial annotations all match, the recognition net is able to extract phase field values from video frames and the neural differential equation net captures the correct phase-field parameters.

Recognition Net. The recognition net predicts the three field variables c_v, c_i and η from in-situ experiment video frames. Under a transmission electron microscope (TEM), void clusters, or the η variable, can be reliably observed (see Figure 2 for void clusters in the actual TEM pictures). The void and interstitial defect percentages (c_v and c_i) are depicted as black shades but cannot be reliably observed due to noise caused by small perturbations, e.g., slight bending of the material samples. The bending of samples is in the scale of nanometers, which cannot be eliminated experimentally, even given the best effort. Therefore, we treat c_v and c_i as unobserved variables.

The η variable can be predicted mainly from the video frames by the recognition net, i.e., $\eta(.,t) = RN_\eta(V_t)$. As a way to estimate hidden variables c_v and c_i, we introduce location embedding vectors into our recognition net model. Let $l_1, \ldots, l_N$ be N vectors, where l_t is the location embedding vector for time t. The value of these vectors vary continuously and slowly with time t. Our first idea was to build the recognition net for c_v and c_i as $c_v(.,t) = RN_v(l_t)$ $c_i(.,t) = RN_i(l_t)$. Here, RN_v and RN_i are two neural nets which translate the location embedding vector l_t into the field variables c_v and c_i at time t, which are both matrices of the size $N_x \times N_y$. As a second idea, we also include the video frame at time t, V_t, as the input, since it offers partial information (the black shades). As the final result, the three field variables are predicted from an uniform architecture $c_v(.,t) = RN_v(l_t, V_t)$, $c_i(.,t) = RN_i(l_t, V_t)$, and $\eta(.,t) = RN_\eta(l_t, V_t)$.

In practice, the three recognition nets, $RN_v(l_t, v_t)$, $RN_i(l_t, v_t)$, $RN_\eta(l_t, v_t)$ are all implemented using the UNet architecture [35]. UNet follows a contracting then expanding neural path. Our motivation for using UNet as the recognition net stems from its wide use in scientific community, although in principle any pixelwise pattern recognition network can be used in this case. In our implementation, the input of the UNet are the video frames V_t. The location embedding vectors l_t are appended to the bottleneck vector in UNet.

Neural Phase-field Net. One of our key contributions is to encode a finite difference phase field model as a differential equation network. As a result, the neural differential equation net can be embedded in the overall neural network architecture, allowing end-to-end training. The high-level idea is to use finite difference to approximate the Cahn-Hilliard and Allen-Cahn equations. We

present here the details of embedding the Cahn-Hilliard equation. Similar process applies for the Allen-Cahn equation. Recall the Cahn-Hilliard equation is as follows:

$$\frac{\partial u}{\partial t} = \nabla \cdot \left(M \nabla \frac{1}{N} \frac{\delta F}{\delta u} \right) = \frac{M}{N} \nabla^2 \left(\frac{\delta F}{\delta u} \right).$$

Using finite difference approach as described previously, especially noting $\nabla^2 f$ can be approximated by the five point stencil centered second-order difference, $\nabla^2 f_{i,j,k} = \frac{1}{\mathrm{d}s^2} \left(f_{i+1,j,k} + f_{i-1,j,k} + f_{i,j+1,k} + f_{i,j-1,k} - 4 f_{i,j,k} \right)$, the Cahn-Hilliard equation can be written as:

$$u_{k+1} = u_k + \frac{M}{N} \frac{dt}{\mathrm{d}s^2} Conv(\frac{\delta F}{\delta u}, K). \tag{3}$$

Here, u_k is a discretized matrix of field variable u in which the i, j-th entry of u_k is $u(x_i, y_j, t_k)$. The functional derivative $\frac{\delta F}{\delta u}$ is also a matrix, whose i, j-th entry is $\frac{\delta F}{\delta u}(x_i, y_j, t_k)$. $\frac{\delta F}{\delta u}$ can be derived by hand. $Conv$ means to convolve $\frac{\delta F}{\delta u}$ with kernel K, where

$$K = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

Equation 3 gives out a finite difference form to obtain the value of the field variable u_{k+1} in the next time stamp from the current value u_k. Interestingly, the temporal dynamics of u_k can be calculated via a convolutional operator, which can be implemented as a neural network layer relatively easily and subsequently embedded into `NeuraDiff`. Notice that a key difference between our neural differential equation net and a common convolutional layer is that, the convolution kernel is learned through training in a classical convolutional net. However, in our neural differential equation net, we keep the convolution kernel fixed, and learn the parameters associated with the variables in free energy F.

Overall Architecture and Training. The overall architecture of `NeuraDiff` combines the recognition net with the neural differential equation net. We arrange the dataset into pairs of frames which are T time stamps apart: $\mathcal{D} = \{(t_i, V_{t_i}, A_{t_i}, V_{t_i+T}, A_{t_i+T}) \mid i = 1, \ldots, M\}$. Here, V_{t_i} is the video frame at the time stamp t_i. A_{t_i} is the annotation for V_{t_i}. V_{t_i+T} and A_{t_i+T} are the video frame and its annotations at the time stamp $t_i + T$. First, V_{t_i} are fed into the recognition net together with the location embedding l_{t_i} to produce the predicted field variables c_v^*, c_i^*, and η^* at time stamp t_i. We replace the portion of η^* with the ground-truth annotation if the annotation is available. The updated value of η^* is denoted by η^{**}. After this update, c_v^*, c_i^*, and η^{**} values are sent to the neural differential equation net to simulate for T steps. The results of the simulation are $c_v^*(t_i + T)$, $c_i^*(t_i + T)$, and $\eta^*(t_i + T)$. At $t_i + T$, the recognition net produces the recognized field variables $\hat{c}_v(t_i + T)$, $\hat{c}_i(t_i + T)$ and $\hat{\eta}(t_i + T)$. Along with the annotations, the triage loss function that the neural network model optimizes, penalizes three types of mismatches:

$$\mathcal{L} = L_{sim} + \lambda_1 L_{rec} + \lambda_2 L_{sim-rec}.$$

Here, L_{sim} denotes the loss function for the mismatch between simulated η^* and the annotations A: $L_{sim} = \|\mathbf{1}_A(\eta^* - A)\|^2$. $\mathbf{1}_A$ is the indicator matrix

of annotations, the entry of which is 1 if the corresponding entry in A is not $*$. L_{rec} denotes the loss penalizing the mismatch between recognized $\hat{\eta}$ and the annotations A, $L_{rec} = \|\mathbf{1}_A(\hat{\eta} - A)\|^2$. $L_{sim-rec}$ denotes the penalties between the simulated and recognized phase-field variables $L_{sim-rec} = \left(\|\eta^* - \hat{\eta}\|^2 + \|c_v^* - \hat{c_v}\|^2 + \|c_i^* - \hat{c_i}\|^2\right)$ In these equations, all phase-field variables are at time $t_i + T$. λ_1 and λ_2 are hyper-parameters that balance the relative importance of terms. The entire neural network structure is trained via stochastic gradient descent. A minibatch of frames are sampled for the back-propagation algorithm in each iteration.

5 Related Work

AI Driven Scientific Discovery There has been a recent trend to leverage AI for scientific discovery. In materials science, CRYSTAL is a multi-agent AI system to solve the phase-map identification problem in high-throughput materials discovery [15]. Neural models have been proposed to generate optimized molecule designs; see, e.g., the Attentive Multi-view Graph Auto-Encoders [28], the Junction Tree Variational Autoencoder [19, 20], message passing neural networks [33]. Attia et al. demonstrate a machine learning methodology to efficiently optimize the parameter space for fast-charging protocols in electric-vehicles [3]. Bayesian optimization and reinforcement learning have also been used in budgeted experimental designs [4]. Embedding physics knowledge in machine learning has also attracted attention. The work by [27] adds to variational autoencoders constraints as regularization terms to improve the validity of the molecules generated. Grammar Variational Autoencoders [23] provided generative modeling of molecular structures by encoding and decoding directly to and from these parse trees, ensuring their validity. Constraint driven approaches such as satisfiability modulo theory (SMT) have also been used to ensure physically meaningful results [12]. In addition, the work of [38] proposed a new supervising approach to learn from physical constraints.

Embedding Optimization in Neural Architectures Amos et al. proposed to embed quadratic program as a layer in an end-to-end deep neural network [2]. Recently, Ferber et. al. proposes to embed a mixed integer program as neural network layers [13]. Devulapalli et al. proposed a neural network capable of back-propagating gradients through the matrix inverse in an end-to-end approach for learning a random walk model [11]. Dai et al. proposed learning good heuristics or approximation algorithms for NP-hard combinatorial optimization problems to replace specialized knowledge and trial-and-error [21]. The work by [18] proposes a new programming language for differentiable physical simulation. [37] proposes a graph network based simulator, where a stack of embedded graph networks in an encoder-decoder architecture is used to learn the dynamics of particles interacting in a 3D environment.

Learning PDEs Previous work has discovered approaches that include physics information in machine learning, where the physics models are represented by differential equations; see, e.g., in turbulence prediction [32]. Bezenac et al. used

a convolutional-deconvolutional (CDNN) module to predict the the motion field from a sequence of past images for sea surface temperature forecasting, motivated by the solution of a general class of partial differential equations [6]. Lutter et al. proposed Deep Lagrangian Networks that can learn the equations of motion of a mechanical system with a deep network efficiently while ensuring physical plausibility. Their approach incorporates the structure introduced by the ODE of the Lagrangian mechanics into the learning problem and learns the parameters in an end-to-end fashion [26]. Time-aware RNNs [10] utilized the similarities between a set of discretized differential equations and the RNN network to model the system equations from a physics system. PDE-Net [24] was proposed to accurately predict dynamics of complex systems by representing the PDEs with convolutional networks where all filters are properly constrained. Neural ODE was introduced in [8, 25], where the output of a neural network is treated as the continuous-time derivative of input, thus providing an interface to incorporate differential equation modeling into machine learning models. Their work have inspired a number of other ideas. For example [22] uses a natural spline to handle irregularly observed time series data with Neural CDE model. While the original Neural ODE model was designed for continuous time modeling, discrete time modeling have been proposed as well by [29]. Hamiltonian neural network (HNN) as proposed in [16] uses partial derivatives of the final output instead of the actual output value, to approximate an energy function and build a Hamiltonian system with a neural network. To make learning easier in HNN, [14] propose a change in system representation along with explicit constraints. A separate line of work [17, 5] exploit this connection to solve PDEs. Most of these works learn PDEs from the observed trajectories, which in many applications need to be extracted in separate steps. For example, the TEM videos in our application only provide partial information on the actual trajectories of the phase-field variables. Our `NeuraDiff` integrates a computer vision neural network with a PDE neural net in the discovery of physics models directly from experiment data.

Image Analysis for In-situ Data Automated image segmentation models are being developed to identify defects and other nanostructures in TEM images [34]. However, they do not learn any physics models.

6 Experiments

Our experiments on both the synthetic and real-world datasets demonstrate that our `NeuraDiff` provides highly accurate tracking while at the same time learns the phase-field model that governs nanostructure dynamics.

Training. In the experiments, we first pre-train the recognition net before the entire architecture, to predict the 3 phase-field variables using only video frames. The details of this pre-training step is provided in the supplementary materials. The stochastic optimization algorithm we used for both pre-training and the actual training is Adam, with the initial learning rate set to be 0.01. Additional details on the experiment setup, train-test split and hyperparameter tuning are in the supplementary materials.

Accuracy	NeuraDiff	UNet Baseline
Synthetic Data	98.5%	99.9%
Real Data	96.2%	96.4%

Table 1. Our NeuraDiff obtains similar and near perfect tracking accuracy as a UNet baseline in both the synthetic and the real-world datasets.

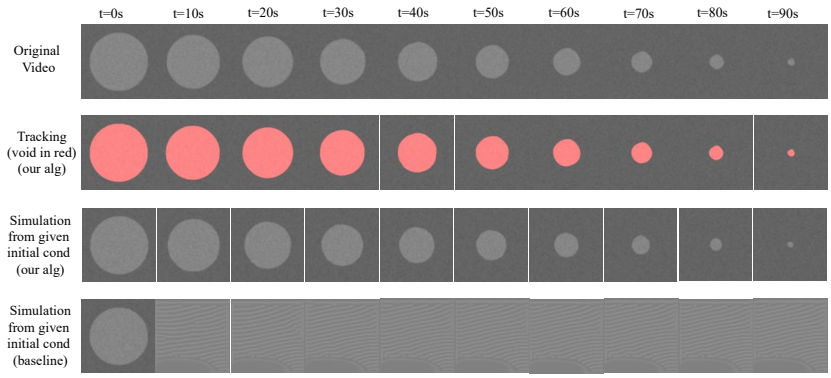


Fig. 4. Our NeuraDiff provides reliable tracking (2nd row, the tracking result of the voids shown in red) while learning the correct physics on synthetic data. In the third row, we simulate our NeuraDiff with the learned parameters from a given initial condition. The learned model simulates a similar dynamics as the original video. Nevertheless, a neural network baseline without embedding the phase-field model produces unsatisfying result (4th row).

Dataset Description. We use both synthetic data and real-world in-situ experiment data to evaluate our model. Both the synthetic and real-world data are in high frame rate high resolution video format. For generating synthetic video data, we use the void evolution model as described in [30]. For real-world data, we use the in-situ experiment video showing the evolution of void defects in Cu 110, as captured through Transmission Electron Microscopy (TEM) imaging. The details of synthetic data generation process, the testbed conditions during in-situ radiation experiments and annotation process for in-situ experiment data are provided in the supplementary materials.

Highly Accurate Tracking Accuracy. Our NeuraDiff provides highly accurate tracking accuracy, together with a UNet baseline, which were trained to predict the η values from the video frames using supervised learning. From the phase-field model, η varies continuously, is close to 1.0 within the void cluster and is close to 0.0 outside. However, the annotation matrix A is binary (1 for void cluster and 0 for others). We cut off η values at 0.5 and evaluate the accuracy in the following way: $\frac{1}{N_x N_y} \sum_{x,y} \mathbb{1}(\eta_{x,y} \geq 0.5, A_{x,y} = 1) + \mathbb{1}(\eta_{x,y} < 0.5, A_{x,y} = 0)$.

We can see from table 1 that both our NeuraDiff and the UNet baseline produce close to optimum tracking results. The second row of Figure 4 and Figure 5 depict the actual tracking of void clusters on synthetic data as well as

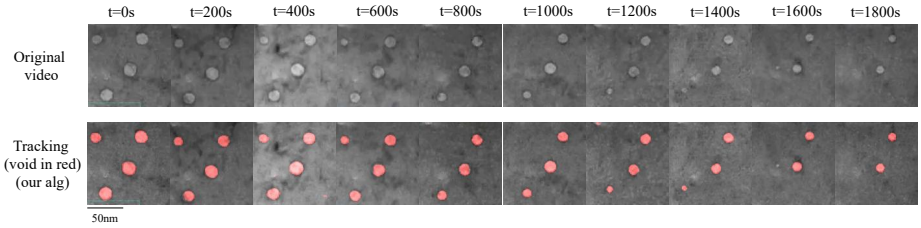


Fig. 5. Our `NeuraDiff` provides accurate tracking (in red) on real in-situ experiment of Cu at the temperature of $350^{\circ}C$ and $0.25 - 1.00$ dpa of irradiation.

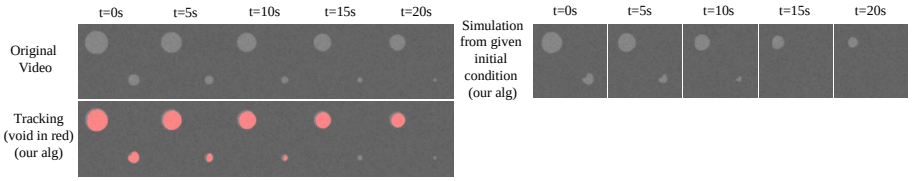


Fig. 6. Transfer learning result for `NeuraDiff`. It provides reliable tracking (second row, in red) and predict correct void progression on unseen data. `NeuraDiff` was trained on the dataset used in Figure 4 including a single nanovoid, and was not fine tuned when evaluated on this dataset of several nanovoids.

on real experimental data. The region of void clusters is highlighted with the red color. We can visually inspect that the tracking is close to optimum.

Capture the Physics. Aside from providing accurate and reliable tracking, our `NeuraDiff` also learns the phase-field model that correctly predicts void cluster evolution. Our key contribution lies within the fact that our model can learn the dynamics of void evolution without compromising the tracking task, and needs less data for the tracking purpose. With the learned parameters, we simulate the evolution of the phase-field variables from the initial condition of the synthetic dataset using finite difference. The initial condition is given as the first frame of the video, instead of the values of the three field variables. Our model has to infer their values from the recoginition net. The result is shown in the third row of Figure 4. We can see that the dynamics closely resembles that of the original video, suggesting that our approach identifies the correct phase-field model. We point out that the learned parameters as well as the predicted c_v and c_i unobserved field variables are different from the original values used to synthesize the dataset. This suggests that there are multiple parameter values which lead to similar dynamics. We also evaluated our model performance in transfer learning. Here, the model was trained on the synthetic dataset involving one void, but was tested for both tracking and void evolution in an unseen dataset involving multiple void of different sizes (figure 6). Our model produces reasonable tracking results and simulates the correct dynamics. See more details in the supplementary materials.

We tried hard to use a neural network model to predict void evolution without embedding the phase-field model. However, the result is not satisfying. For example, in the fourth row of Figure 4, we used the UNet to predict the next

frame given the current frame. Then we use the UNet to synthesize the entire video via repeated predictions of the next frame. However, the performance is not satisfying as the noise quickly dominates the signals. We even tried to feed the neural network with the correct values of the three field variables and ask it to predict the next frame. Note the field variable values are not available in real-world experiments. The baseline neural network cannot predict the dynamics even with these additional inputs.

7 Conclusion

We present `NeuraDiff`, an end-to-end model to learn a physics model characterized by a set of partial differential equations directly from data. Our key idea is to embed the physics model as a multi-layer convolutional neural net into the overall neural architecture for end-to-end training. We apply `NeuraDiff` in the task of tracking and characterizing the dynamics of point defect clusters in materials under high temperatures and heavy irradiations. Our approach produces near perfect tracking and is able to capture a physics model that predicts future nanostructure dynamics correctly, which are not possible for pure data-driven machine learning models. Our model is validated on both synthetic and real experimental data. Future work include to scale up the computation for high dimensional, high frame rate videos, and to validate the physics models learned from our framework with more real-world irradiation experiments.

Acknowledgements

This research was supported by NSF grants IIS-1850243, CCF-1918327. We thank anonymous reviewers for their comments and suggestions.

References

1. Allen, S.M., Cahn, J.W.: Ground state structures in ordered binary alloys with second neighbor interactions. Acta Metallurgica **20**(3), 423–433 (1972)
2. Amos, B., Kolter, J.Z.: Optnet: Differentiable optimization as a layer in neural networks. In: International Conference on Machine Learning. pp. 136–145 (2017)
3. Attia, P.M., Grover, A., Jin, N., Severson, K.A., Markov, T.M., Liao, Y.H., Chen, M.H., Cheong, B., Perkins, N., Yang, Z., et al.: Closed-loop optimization of fast-charging protocols for batteries with machine learning. Nature **578**(7795), 397–402 (2020)
4. Azimi, J., Fern, X.Z., Fern, A.: Budgeted optimization with constrained experiments. J. Artif. Int. Res. **56**(1), 119–152 (May 2016)
5. Beck, C., E, W., Jentzen, A.: 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 (2019)
6. de Bezenac, E., Pajot, A., Gallinari, P.: Deep learning for physical processes: Incorporating prior scientific knowledge. In: International Conference on Learning Representations (2018)

7. Cahn, J.W., Hilliard, J.E.: Free energy of a nonuniform system. i. interfacial free energy. The Journal of chemical physics **28**(2), 258–267 (1958)
8. Chen, R.T.Q., Rubanova, Y., Bettencourt, J., Duvenaud, D.K.: Neural ordinary differential equations. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 31, pp. 6571–6583 (2018)
9. Chen, Z., Zhang, J., Arjovsky, M., Bottou, L.: Symplectic recurrent neural networks. In: 8th International Conference on Learning Representations, ICLR (2020)
10. Demeester, T.: System identification with time-aware neural sequence models. arXiv preprint arXiv:1911.09431 (2019)
11. Devulapalli, P., Dilkina, B., Xue, Y.: Embedding conjugate gradient in learning random walks for landscape connectivity modeling in conservation. In: Bessiere, C. (ed.) Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20. pp. 4338–4344. International Joint Conferences on Artificial Intelligence Organization (7 2020)
12. Ermon, S., Le Bras, R., Gomes, C.P., Selman, B., Van Dover, R.B.: Smt-aided combinatorial materials discovery. In: International Conference on Theory and Applications of Satisfiability Testing. pp. 172–185. Springer (2012)
13. Ferber, A., Wilder, B., Dilkina, B., Tambe, M.: Mipaal: Mixed integer program as a layer. In: AAAI. pp. 1504–1511 (2020)
14. Finzi, M., Wang, K.A., Wilson, A.G.: Simplifying hamiltonian and lagrangian neural networks via explicit constraints. Advances in Neural Information Processing Systems **33** (2020)
15. Gomes, C.P., Bai, J., Xue, Y., Björck, J., Rappazzo, B., Ament, S., Bernstein, R., Kong, S., Suram, S.K., van Dover, R.B., et al.: Crystal: a multi-agent ai system for automated mapping of materials' crystal structures. MRS Communications **9**(2) (2019)
16. Greydanus, S., Dzamba, M., Yosinski, J.: Hamiltonian neural networks. Advances in Neural Information Processing Systems **32**, 15379–15389 (2019)
17. Han, J., Jentzen, A., E, W.: Solving high-dimensional partial differential equations using deep learning. Proceedings of the National Academy of Sciences **115**(34), 8505–8510 (2018)
18. Hu, Y., Anderson, L., Li, T., Sun, Q., Carr, N., Ragan-Kelley, J., Durand, F.: Difftaichi: Differentiable programming for physical simulation. In: 8th International Conference on Learning Representations, ICLR (2020)
19. Jin, W., Barzilay, R., Jaakkola, T.: Junction tree variational autoencoder for molecular graph generation. In: International Conference on Machine Learning (2018)
20. Jin, W., Yang, K., Barzilay, R., Jaakkola, T.: Learning multimodal graph-to-graph translation for molecule optimization. In: International Conference on Learning Representations (2018)
21. Khalil, E., Dai, H., Zhang, Y., Dilkina, B., Song, L.: Learning combinatorial optimization algorithms over graphs. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 6348–6358 (2017)
22. Kidger, P., Morrill, J., Foster, J., Lyons, T.: Neural Controlled Differential Equations for Irregular Time Series. arXiv:2005.08926 (2020)
23. Kusner, M.J., Paige, B., Hernández-Lobato, J.M.: Grammar variational autoencoder. In: International Conference on Machine Learning. pp. 1945–1954 (2017)
24. Long, Z., Lu, Y., Ma, X., Dong, B.: Pde-net: Learning pdes from data. In: International Conference on Machine Learning. pp. 3208–3216 (2018)

25. Lu, Y., Zhong, A., Li, Q., Dong, B.: Beyond finite layer neural networks: Bridging deep architectures and numerical differential equations. In: International Conference on Machine Learning. pp. 3276–3285 (2018)
26. Lutter, M., Ritter, C., Peters, J.: Deep lagrangian networks: Using physics as model prior for deep learning. In: International Conference on Learning Representations (2018)
27. Ma, T., Chen, J., Xiao, C.: Constrained generation of semantically valid graphs via regularizing variational autoencoders. In: Advances in Neural Information Processing Systems. pp. 7113–7124 (2018)
28. Ma, T., Xiao, C., Zhou, J., Wang, F.: Drug similarity integration through attentive multi-view graph auto-encoders. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI. pp. 3477–3483 (7 2018)
29. Matsubara, T., Ishikawa, A., Yaguchi, T.: Deep energy-based modeling of discrete-time physics. In: Advances in Neural Information Processing Systems 33 (NeurIPS2020) (2020)
30. Millett, P.C., El-Azab, A., Rokkam, S., Tonks, M., Wolf, D.: Phase-field simulation of irradiated metals: Part i: Void kinetics. Computational materials science **50**(3), 949–959 (2011)
31. Niu, T., Nasim, M., Annadanam, R., Fan, C., Li, J., Shang, Z., Xue, Y., El-Azab, A., Wang, H., Zhang, X.: Recent studies on void shrinkage in metallic materials subjected to in situ heavy ion irradiations. JOM **72** (09 2020). https://doi.org/10.1007/s11837-020-04358-3
32. Portwood, G.D., Mitra, P.P., Ribeiro, M.D., Nguyen, T.M., Nadiga, B.T., Saenz, J.A., Chertkov, M., Garg, A., Anandkumar, A., Dengel, A., et al.: Turbulence forecasting via neural ode. arXiv preprint arXiv:1911.05180 (2019)
33. Raza, A., Sturluson, A., Simon, C.M., Fern, X.: Message passing neural networks for partial charge assignment to metal–organic frameworks. The Journal of Physical Chemistry C **124**(35), 19070–19082 (2020)
34. Roberts, G., Haile, S.Y., Sainju, R., Edwards, D.J., Hutchinson, B., Zhu, Y.: Deep learning for semantic segmentation of defects in advanced stem images of steels. Scientific reports **9**(1), 1–12 (2019)
35. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
36. Sæmundsson, S., Terenin, A., Hofmann, K., Deisenroth, M.P.: Variational integrator networks for physically structured embeddings. In: The 23rd International Conference on Artificial Intelligence and Statistics. vol. 108, pp. 3078–3087 (2020)
37. Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J., Battaglia, P.W.: Learning to simulate complex physics with graph networks. In: International Conference on Machine Learning (2020)
38. Stewart, R., Ermon, S.: Label-free supervision of neural networks with physics and domain knowledge. In: 31 AAAI Conference on Artificial Intelligence (2017)
39. Tong, Y., Xiong, S., He, X., Pan, G., Zhu, B.: Symplectic neural networks in taylor series form for hamiltonian systems. ArXiv **abs/2005.04986** (2020)
40. Zhong, Y.D., Dey, B., Chakraborty, A.: Symplectic ode-net: Learning hamiltonian dynamics with control. In: 8th International Conference on Learning Representations, ICLR (2020)