

Multi-input convolutional network for ultra-fast simulation of field evolution

Zhuo Wang¹, Wenhua Yang², Lingyan Xiang¹, Xiao Wang³, Yingjie Zhao⁴,
Yaohong Xiao¹, Pengwei Liu⁴, Yucheng Liu², Zikanov Oleg¹, Lei Chen^{1*}

¹ Department of Mechanical Engineering, University of Michigan-Dearborn, Dearborn, MI 48128, USA

² Department of Mechanical Engineering, Mississippi State University, Starkville, MS 39762, USA

³ School of Mechatronic Engineering, Jiangsu Normal University, Xuzhou, Jiangsu 221116, China

⁴ College of Mechanical Engineering, Yanshan University, Qinhuangdao, Hebei 066004, China

Abstract

There is a compelling need for the capability of multi-input field-to-field regression or mapping the initial field and applied conditions to the evolved field, e.g., given current flow field and fluid properties predicting next-step flow field. Such a regression capability can provide a maximum substitute of various physics-based models, enabling fast simulation of condition-dependent field evolvments. We hereby propose a y-shaped multi-input convolutional network (ConvNet), yNet, which can efficiently build functional relationship between multi-input (i.e., input field and applied condition parameters) and evolved field. yNet merges input signals without flattening the field input into a large vector and, instead, by directly manipulating high-level encoded feature maps. It thus possesses extreme computational efficiency (i.e., as much as six orders of magnitude faster) against physics-based models, as well as significant model reduction (i.e., one-tenth of the model size) over its ConvNet counterpart. It is applied to solving a variety of data-driven modeling problems in physical science and engineering, i.e., fluid dynamics, porosity evolution in selective laser sintering (SLS), and stress field development. yNet consistently shows great extrapolation capability in predicting beyond the training datasets in terms of temporal ranges, spatial domains, and geometrical shapes. The three applications have demonstrated the general effectiveness of yNet in learning the underlying physical rules that govern field evolution and development. The well-tested yNet thus may have an enormous potential and profound impact as a conceptually simple, light-weight yet powerful multi-input field-to-field regressor.

Keywords: Convolutional Network, Scientific Machine Learning, Data-driven Modeling

* leichen@umich.edu (L. Chen).

Introduction

Deep convolutional network (ConvNet) utilizes stacks of convolutional layers for automatic, hierarchical representation learning of high-dimensional data. The convolutional components of ConvNet bring it great advantages in processing data in the form of multiple arrays, such as signals and sequences (1D), images or audio spectrograms (2D) and video or volumetric images (3D) [1]. Since its first application in digit recognition nearly 30 years ago [2], ConvNet has been particularly serving as the backbone for performing various computer vision (CV) and other image-centered tasks, such as image labeling/classification [3], object localization [4], semantic segmentation (pixel-wise labeling) [5, 6], etc. Upon realization of its distinct advantage in image modeling, it has been recently witnessed a surge of applications in different scientific domains where image-involved problems are ubiquitous. Some of those scientific applications include image classification with respect to scientific images [7, 8], microstructure characterization and reconstruction (MCR) [9-11], process-structure [12, 13] and structure-property [14, 15] relationship modeling in materials science and engineering. They all leverage ConvNets to explicitly process and “understand” scientific images, thereby free from hand-craft featurization and with minimal human intervention. For example, conventional structure-property modeling relies heavily on domain experts to design effective structure descriptors [16, 17], while ConvNets directly take raw structure information as input and can construct high-fidelity relationship between image-based structure and property; see Fig. 1a and 1b respectively.

Another prominent application is ConvNet based simulation of field evolvment and development. This application also takes advantage of its superior image modeling

capability because different types of physical fields, such as temperature, velocity and density fields, can eventually present as images. In this regard, ConvNets treat the field evolution and development as a purely data-driven, image-to-image/field-to-field regression problem; see Fig. 1c. Some typical examples include fluid flow evolution (i.e., current flow field to future flow field [18]), microstructure evolution (i.e., current structural field to future structural field [19]) and stress/strain field development (structural field to stress/strain field [20, 21]). ConvNets simply aim to build those field-to-field mappings from training datasets, without ever playing with any complex physical governing equation. As such, the ConvNet, once trained, can offer an inexpensive substitute of physics-based model, enabling fast yet realistic simulation of field evolution and development.

However, most of the related research [19, 21-27] just adopted the CV task oriented ConvNets, especially those with a typical encoder-decoder architecture. Obviously, such an encoder-decoder model originally for semantic segmentation is unable to satisfactorily address data-driven modeling of field evolutions. In semantic segmentation, the output image or segmentation result is uniquely determined by the input image to be segmented; see Fig. 1c. However, the evolution and development of a physical field is usually not only decided by the initial field, but also controlled by the applied conditions. For instance, flow field developed at a future point is concerned with the current flow field as well as fluid properties (see Fig. 1d) and stress field development greatly depends on the condition of applied force. With a pure image-to-image regression tool, those existing researches have to fix conditions, thus failing to incorporate condition-related parameters. Regarding this, the trained ConvNet is strictly applicable to

the specific condition used for training. This is apparently contrary to the general fact of great variability in applied conditions, and seriously stifles the practical use of the trained ConvNet. Thus, developing a more general multi-input field-to-field regressor, which can build functional relationship between multi-input (i.e., input field and condition parameters) and output field, would be of huge significance. It will, for example, allow us to train a single ConvNet applicable to various fluids with distinct conditions (Fig. 1d), instead of training separate ConvNet for each of them. Such a multi-input field-to-field/image-to-image regression capability with well-tested general usefulness is fundamentally important, as are the other three regression capabilities. It will solidly advance the border of our reachable regression tasks, as compared between Fig. 1c and Fig. 1d.

Towards this end, we propose a multi-input deep convolutional neural network, which naturally exhibits a signature y-shape as a multi-input field-to-field regressor; see Fig. 1f. We thus name it “yNet” that conveys its most salient feature, namely the efficient fusion of an additional input signal of condition-related parameters, in comparison with the pure image-to-image regression neural network (thus I-shaped) in aforementioned CV and scientific applications (Fig. 1c). As shown, the main architecture is basically composed of encoder-decoder based on deep convolutional network and a branch of multilayer perceptron (MLP). The encoder essentially plays a role of non-linear dimension reduction. It decomposes the original high-dimensional pixel-based field into N_2 information-rich $n_1 \times n_2$ feature maps; see Fig. 1g. MLP acts to expand the condition-related parameter(s) into N_1 neurons that generate a $1 \times N_1$ embedding vector. In this way, the encoder and MLP would facilitate effective fusion of input signals at their ends, as

further discussed later. Decoder serves to correctly reconstruct the merged signal back to a meaningful and desired output field through the deep deconvolution process. Note that, in the deep deconvolution process we also concatenate feature maps extracted during the early autoencoding process. This technique is commonly adopted in semantic segmentation networks, such as U-Net [28] and FCN [29], with the aim of improving segmentation details. It is expected to compensate information loss by coarsening during max-pooling in encoding process and improve information flow from (image) input to decoder.

After building the main architecture, merging signals of multiple inputs poses as a key problem. In fact, such multiple mixed inputs (a high-dimensional image and scalar parameters) are not uncommon in various ML tasks, although the output is not necessarily image. However, the merging strategies in existing multi-input ConvNets can suffer from various drawbacks associated with dense parameters. One common merging strategy [30-32] is to flatten image signal, here represented by its feature maps, into a long vector, thus making it compatible and hence concatenable to the vectorial output of MLP; see Fig. 1e. Another merging strategy in image captioning model [33], which is not graphically illustrated here as it is very similar to the above one, is also flattening image first, however, strictly to a vector of the same size (i.e., $N_1 = N_2$). In doing so, it allows for element-wise adding of two vectors (image and linguistic vectors) to successfully merge signals. Those flattening-based strategies (flattening+concatenating or flattening+adding) are suitable or sometimes unavoidable for predicting a single label or numeric value but may not be optimal for ConvNets with an image-type output. The flattening operation would yield large fully connected layers as shown, which are known

to be parameter-intensive. The dense parameters can bring about drawbacks such as increased training complexity requiring sophisticated training schemes, more hardware resources demanded during both training and inference, slow inference speed. The above facts are clearly shown, for example, in comparative studies [34] between the light-weight SegNet and other semantic segmentation ConvNets [29, 35, 36]. Apart from parameter-efficiency issue, flattening operation will interrupt the signal flow of 2D feature maps, which themselves are meaningful as spatial features in condensed and coarse form. The merged signal in those manners however becomes a skinny numerical vector, which is somewhat meaningless or less interpretable. After flattening-based merging, ConvNet would have to take further efforts to force the vectors back to 2D useful maps for decoding process. A flattening-based strategy thus, if workable for multi-input image-to-image regression [37, 38], might make the multi-input ConvNet more like a brute-force regressor.

To overcome the shortcomings of flattening-based merging strategy, we propose to properly merge signals using a one-to-one connection via multiplication (Fig. 1g), which is mathematically akin to the gating mechanism [39]. The proposed gating-based signal merging strategy features great simplicity and would result in smooth signal flow of feature maps throughout the encoding-decoding process. In this case, MLP actually turns as a signal modulator of feature maps, which is anticipated to rearrange encoded high-level feature maps towards high-level representation of the output field. Especially, the effectiveness of the gating-based merging strategy lies in the fact of those neurons interacting with respective feature map in a direct, neat and therefore strong manner. This is a key for yNet to efficiently learn the combined effects of two channels. We posit that,

through training, MLP can learn to precisely manipulate the initial field represented by high-level feature maps into the developed field (preliminarily in the form of gated feature maps) for given condition; the decoder then appropriately reconstruct the gated feature maps into the realistic evolved or developed field.

We further clarify that this research focuses on the parameter-type conditions only. However, the core of yNet should shed light on efficiently incorporating other conditions in more complex forms, as discussed later. That being said, the umbrella term “condition” in this paper refers to not only external conditions (e.g., ambient temperature and applied loading level) but also internal conditions (e.g., materials and physical properties concerned with evolving kinetics), as well as the time period of evolvement. As long as one has an association with the field evolvement and development, it can be seen as “condition”, thus falling within the interest of the proposed yNet.

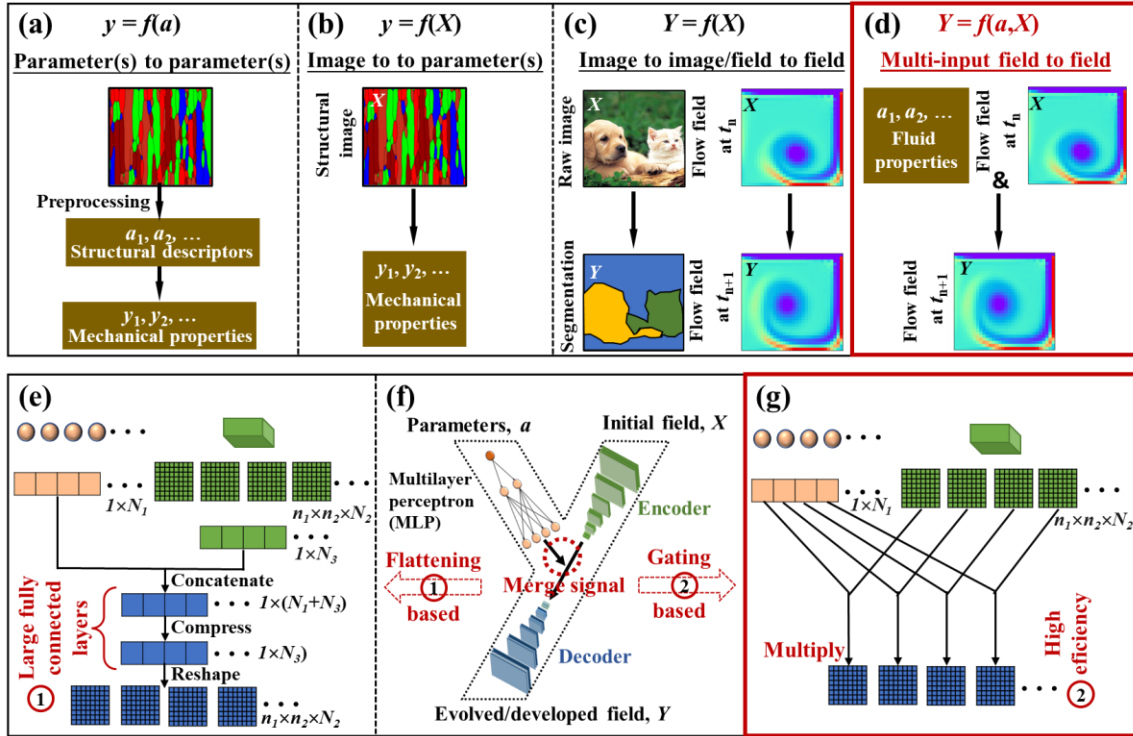


Fig. 1 Four different regression tasks with increasing complexity and multi-input ConvNet. (a) parameter(s) to parameter(s) regression for conventional structure-property modeling, where both the input and output are one or a few parameters, i.e., scalar quantities. (b) image to parameter(s) regression for high-fidelity structure-property modeling, where the input is image and output are parameters. (c) image to image/field to field regression for semantic segmentation and fluid dynamics simulation, where both the input and output are images. (d) multi-input field to field regression for condition-incorporated fluid dynamics simulation, where the mixed input are parameters of fluid properties and image of current flow field and the output is image of next-step flow field. (f) The proposed y-shaped multi-input ConvNet for handling the above multi-input field-to-field regression tasks. Comparison of (e) the conventional flattening-based and (g) the proposed gating-based signal merging strategy.

Next, we will present three applications of yNet to show its different extrapolation capabilities, which collectively demonstrate the general effectiveness of yNet in learning various field evolution and development rules. Following the above three applications, we finally discuss its enormous potential in future applications as well as some limitations.

Application 1: fluid dynamics

In this paper, we start with the widely studied fluid dynamics simulation in the data-driven modeling community. Specifically, we will use a classical fluid flow problem – incompressible, viscous flow shedding around a cylinder. In this case, the spatiotemporal evolution of viscous fluid field is governed by:

$$\frac{\partial \omega}{\partial t} = \nabla \times (\mathbf{v} \times \omega) + \nu \nabla^2 \omega \quad (1)$$

where $\omega = \nabla \times \mathbf{v}$ is the vorticity, $\mathbf{v}$ is the velocity of the fluid, ν is the kinematic viscosity of the fluid. Variables in Eq. (1) can be rewritten in scaled units with respect to unit length L and unit fluid velocity V , namely, $x = x' L$, $\mathbf{v} = \mathbf{v}' V$, $t = t' \frac{L}{V}$, $\omega = \omega' \frac{V}{L}$.

Consequently, Eq. (1) can be rewritten as:

$$\frac{\partial \omega'}{\partial t'} = \nabla' \times (\mathbf{v}' \times \omega') + \frac{1}{Re} \nabla'^2 \omega' \quad (2)$$

where $Re = \frac{LV}{\nu}$ is the Reynolds number as a function of kinematic viscosity, unit length and unit fluid velocity. From Eq. (2), we can find that Reynolds number is the dominating condition that dictates the flow behavior. Therefore, in the following research, we will train a single yNet capable of simulating fluid flow evolution over a range of Reynolds numbers. The trained yNet will be tested on: (1) predicting flow field development for Re conditions unused in the training datasets; (2) simulating dynamic flow field evolution via recursive predictions; (3) temporal extrapolation for predication, i.e., extending the dynamic simulation beyond the original time domain in training data, and (4) computational accuracy and efficiency in comparison with conventional ConvNet.

Training and testing procedure

Fig. 2a illustrates the overall procedure of training an yNet applicable to a range of Reynolds numbers. Vorticity field is selected as flow field of interest and Reynolds number as the condition parameter, but we can easily train yNet for the other flow fields in the same way. To achieve simulation of dynamic field evolution, yNet acts to predict next-step flow field based on the current flow field and Reynolds number, Re . As such,

long-term flow field evolution simulation can be performed by yNet through multi-step recursive predictions as shown in Fig. 2d and Fig. S2a.

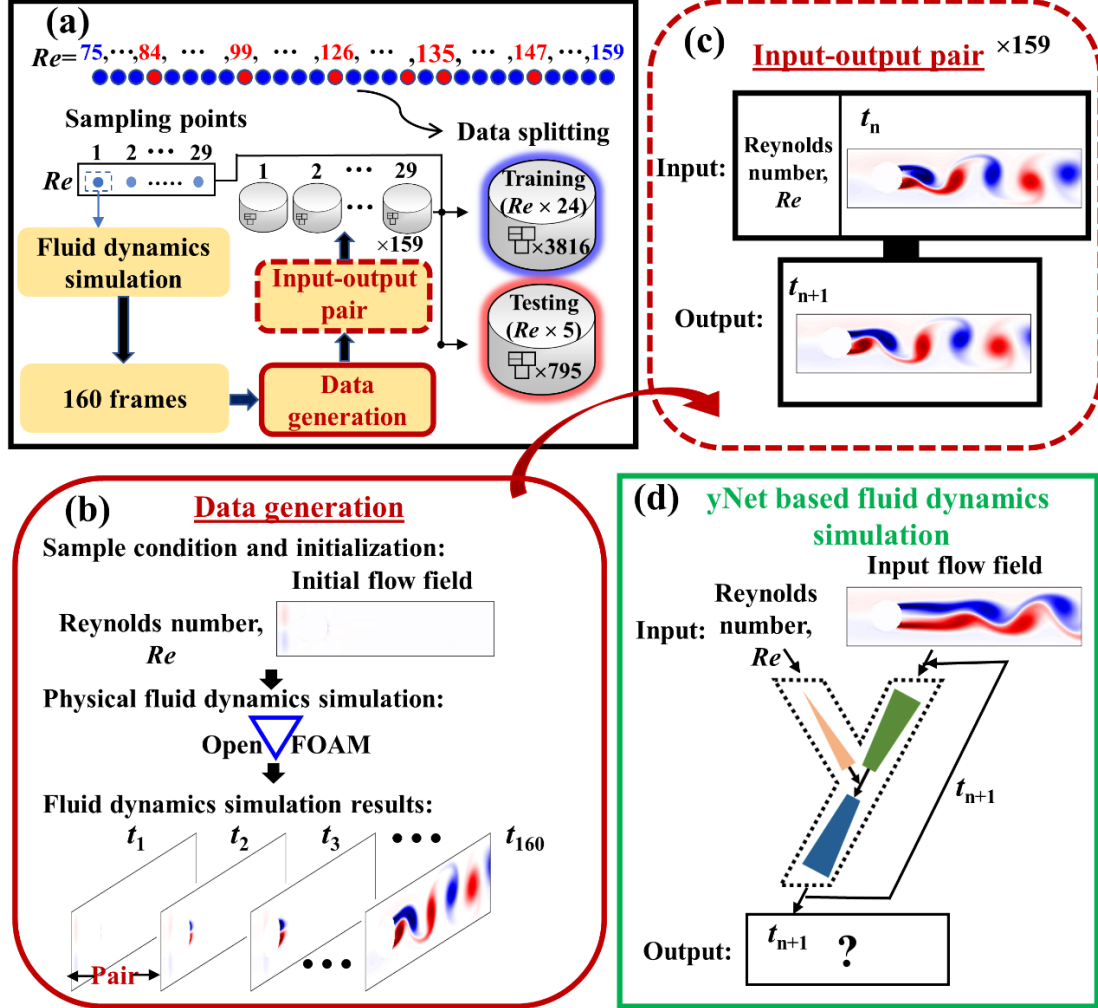


Fig. 2 Illustration of training and testing of yNet for fluid dynamics simulation. (a) Overall flowchart of training and testing yNet. (b) To generate data, a physical fluid dynamics simulation using OpenFOAM is performed at each Reynolds number sampling point, followed by extraction of neighboring snapshots as data pair. (c) Each data point is thus a triplet, consisting of respective Reynolds number, flow fields at t_n and t_{n+1} , where $n = 1, 2, 3, \dots, 159$. (d) The trained yNet can be used to predict flow field for next time step based on given flow field and Reynolds number condition. As such, the trained yNet can also work in a recursive manner to predict long-term flow field evolution.

It should be noted that more condition parameters can be easily incorporated by simple adjustment of input layer of MLP, as executed in next application. To provide dataset, physical simulations are performed at 29 equally spaced Reynolds numbers (i.e., $Re = 75, 78, 81, \dots, 159$), where 24 cases would be randomly chosen for training and the other 5 for testing; see data splitting in Fig. 2a. For all those physical simulations, extraction of neighboring snapshots is performed to generate one-step-distance data pairs, as illustrated in Fig. 2b. A physical simulation will output a total of 160 frames, indicative of 159 data pairs obtained at each Reynolds number sampling point or 4611 for the entire dataset; see Fig. 2c. They are split into training and testing datasets based on Reynolds numbers as mentioned early.

For yNet instantiated in this application, the detailed architecture is illustrated in Fig. S1a and explained in the Method section. We train yNet for 100 epochs with a mini-batch size of 2, by using Adam Optimizer with learning rate = 0.001, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Training process is fully monitored and the weights with the smallest testing loss after reaching convergence are used to generate results in the next subsection.

Testing under unused Re conditions

Fig. 3 presents a set of testing results under the 4 different testing Re conditions. As clearly indicated by the predictive error, yNet can accurately predict the next-step flow field in all cases, with minor difference to the original physic-based simulation. For all the 795 testing results, yNet achieves a negligibly small root mean squared error (RMSE) of 0.588 s^{-2} . It represents one of the first data-driven modeling that directly incorporates Reynolds number. Previous strategies for considering Reynolds number are usually to train a neural network that takes as inputs a series of historical flow fields [18,

40, 41], which as a whole embed corresponding Re information; see LSTM+ConvNet model in Fig. S2b. Thus, the neural network can predict the next step based on the evolving trend of consecutive flow fields, without being informed by any additional inputs like Reynold number. yNet however explicitly incorporates the important condition of Reynolds number, and thus possess two major advantages over previous strategies.

First, yNet is as conceptually simple as a basic ConvNet, without the complexity of integrating sequence-modeling purposed LSTM model [18, 40], thus rendering a significant reduction in model size. Second, yNet is widely applicable to general condition-dependent fluid flow simulation problems. One typical example is to predict the development of flow field for different bluff body shapes [25, 38]. This is essentially a different mapping problem between distinct field quantities, i.e., from structural field (bluff body structure) to flow field; see comparison in Fig. S3. For such regression tasks beyond standard field evolvment, the aforementioned LSTM+ConvNet model apparently will no longer work. We have to explicitly incorporate effects of condition parameters on the flow field development, by developing a general multi-input field-to-field regression capability [38].

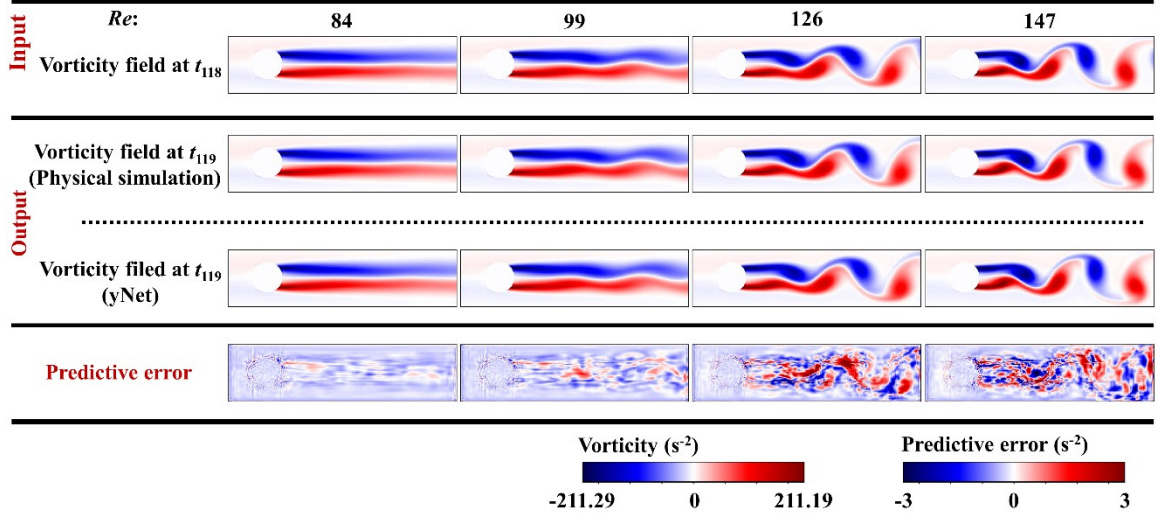


Fig.3 Testing results of yNet under different Re conditions. We randomly pick a set of testing results, which show yNet in predicting next-step flow field development under different Re testing conditions.

Testing on dynamic fluid simulation and temporal extrapolation

Fig. 4a visualizes a representative testing result of predicting multi-step evolution, under the testing condition of $Re = 84$, through recursive inference as depicted earlier in Fig. 2d. Noteworthy is that predictive errors can accumulate after many steps of recursive predictions. However, compared to the ground truth of physics-based simulation, yNet well reproduces the long-term dynamic flow field evolution, suggestive of rather small error during recursive prediction at every step. The predictive error over multi-step predictions for different Re testing conditions is summarized in the left half of Fig. 4c. The quantitative result reveals that predictive error does accumulate as recursive prediction proceeds, but remains within a small level, i.e., $< 12.0 \text{ s}^{-2}$ for different testing conditions.

In addition, attempts have been made to predict fluid dynamics towards the further future, i.e., beyond the time domain in training data; see the visualized example for $Re = 84$ in Fig. 4b. Correspondingly, for validation purpose we also extend the physical simulation for another 32 frames. Despite the big challenge associated with temporal extrapolation, predictive error continues to display gradual accumulation without a sudden rise, as shown in the right half of Fig. 4c. Notably, predictive error can finally reach a maximum RMSE of 29.37 s^{-2} for $Re = 135$. However, such a predictive error is acceptable, still providing good prediction of flow field as clearly shown by the inset in Fig. 4c. The above results indicate the successful learning of underlying evolution dynamics by yNet, which supports its temporal scalability to reasonably predicting flow field evolvement into further future.

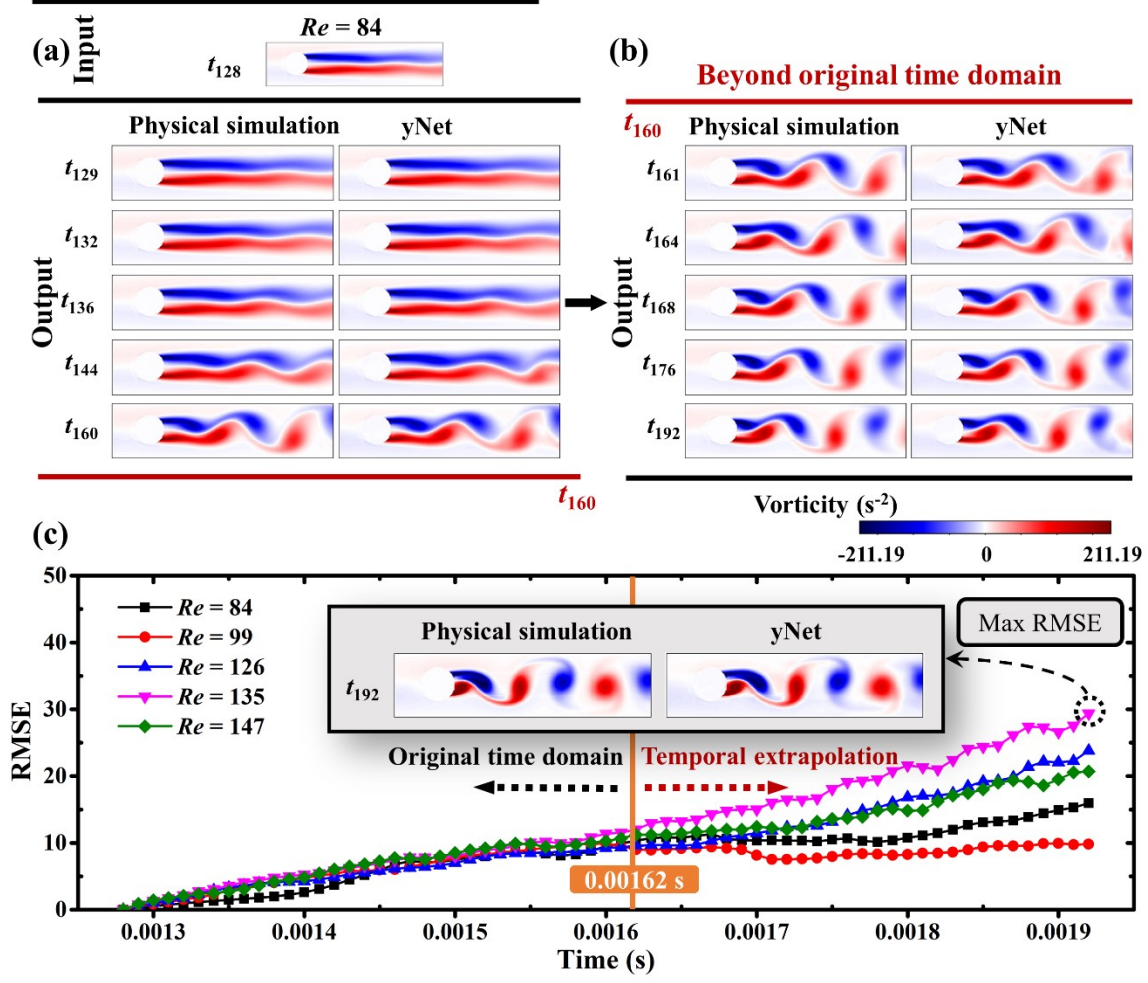


Fig. 4 Testing results of yNet for dynamic fluid simulation. (a) testing of yNet in predicting dynamic flow field evolution with time through recursive predictions. In the dynamic simulation, we start from No. 128 step, which is around the wake of flow under $Re = 84$. We presented snapshots at some time points to quickly show the dynamic evolution process. (b) testing of yNet in predicting dynamic flow field evolution beyond the original time domain in training data. (c) predictive errors over the long-term simulation of fluid dynamics under five Re testing conditions.

Computational efficiency comparison: yNet VS conventional multi-input ConvNet

Fig. 5a and b compares the computational accuracy and efficiency, respectively, between gating-based yNet and the conventional flattening-based multi-input ConvNet. For fair comparison, both networks are implemented with the exact same encoder and

decoder parts, but using gating and flattening based signal-merging strategies respectively. Also, since there are a few variants of flattening based strategy, we implement one by directly concatenating Re to the flattening result. The detailed architectures and configurations of the two implemented ConvNets are graphically compared in Fig. S1. As testing loss curves in Fig. 5a indicate, upon reaching convergence of training, yNet in general achieves comparable modeling accuracy ($MSE = 2.47 \times 10^{-7}$) to that ($MSE = 3.41 \times 10^{-7}$) of the conventional multi-input ConvNet. The inset in Fig. 5a further shows almost the same prediction by the two models at a randomly selected testing data point. However, yNet can substantially reduce the number of model parameters by one order of magnitude and thus enjoys various computational benefits against the conventional ConvNet, such as significantly reduced memory consumption, shorter training time and faster inference speed; see Fig. 5b.

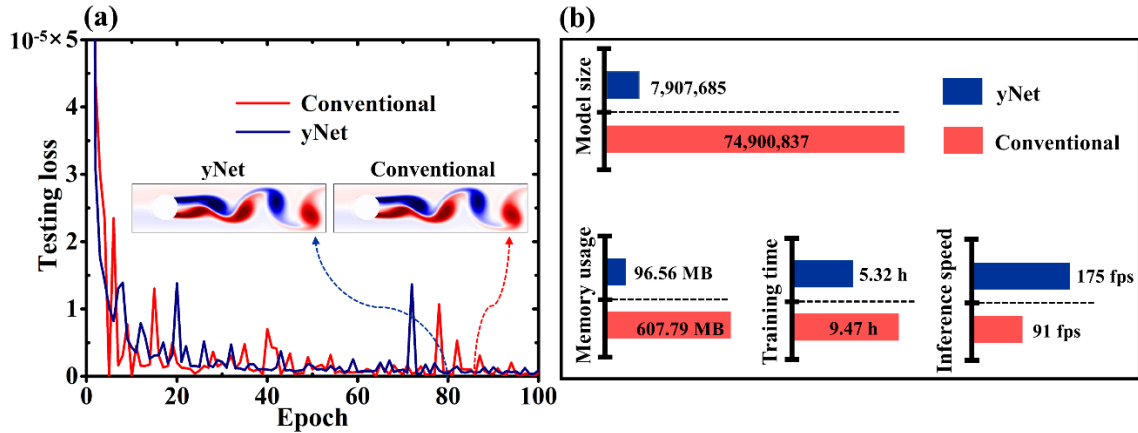


Fig. 5 Comparison of yNet with conventional multi-input ConvNet. (a) testing loss curves of yNet and conventional multi-input CNN; (b) comparison of model size (i.e., total number of parameters) and three other performance metrics between yNet and conventional multi-input CNN. The performance is measured purposely based on a laptop (Intel Core i7-7500U CPU, NVIDIA GeForce GTX 950M GPU, 16G RAM). Note that memory usage is measured with a single unit input, instead of a batch.

The above merits of yNet are attributed to gating-based signal merging strategy, which permits effective interaction between two input channels while inducing only a minimal number of model parameters. The root reason for the effectiveness of directly manipulating feature maps might be further revealed by drawing an analogy to the conventional data compression technique - proper orthogonal decomposition (POD). It is known that POD is close to a basic feedforward neural network with certain constraints [42, 43]. In fact, there are also interesting parallels between POD and ConvNet. POD is a linear-theory based feature extraction technique. Through POD, a high-dimensional image can be expressed as a linear combination of orthogonal modes, i.e., the extracted features. Feature maps in ConvNets, albeit in a 2D matrix form, play a similar role as the coefficients of the linear combination in POD. Like those coefficients corresponding to their respective mode, a feature map in ConvNet would strongly correlate with a certain feature in the pixel space if one projects them back to pixel-based image reversely along the encoding process [44]. The activation values of each feature map are also specifically determined by the input image. The main difference is that the detailed representation of each feature map in high-dimensional image space is not easy to interpret, while those POD coefficients have their own well-decomposed POD mode. However, recent researches prove that one may build separate deconvolution networks to visualize detailed representation of encoded latent variables in high-dimensional space [45] and even analyze their energy contribution like the POD mode decomposition [46]. Although those researches focus on the latent variables (i.e., further compressed latent representation from feature maps), similar methodologies can be used to deconvolve the feature maps to fully reveal their high-dimensional looks. All of those facts imply the

similar function of feature maps in ConvNet to that of coefficients of the linear combination in POD. Thus, direct manipulation of those “coefficients” in ConvNet will effectively generate new combinations of features towards the target image, as does a POD. The deep decoder would then further contribute to the accurate obtainment of target or output image through the non-linear and complex reconstruction.

Application 2: porosity evolution in selective laser sintering

In this section, we further apply yNet to simulation of porosity development in selective laser sintering (SLS) to demonstrate (1) the capability in spatially extrapolating prediction, and (2) extreme computational efficiency over physics-based model. SLS [47] is one of the most popular additive manufacturing (AM) techniques, widely adopted for fabricating metal, ceramic and polymer components in a layer-by-layer manner [48]. Fast prediction of porosity formation in realistically large area with multiple sintering layers is of huge practical significance. However, accurate simulation of porosity formation in SLS usually requires proper consideration of rigid-body translation and rotation of powder particles for particle coalescence, grain growth by boundary migration through various diffusion mechanisms and densification [49]; see Fig. S4. The complicated physics-based SLS simulation is so far computationally restricted to a few layers with limited length, thereby far from SLS practice. There is an urgent need for an inexpensive surrogate of the cumbersome physics-based SLS model.

Training and testing procedure

A multi-physical SLS model [50] is used to generate training and testing data. In the current SLS modeling, the sintering effects (particle coalescence, grain growth and

porosity evolution) are largely decided by the heat-affected zone (HAZ) that depends on the laser condition, and inter-layer interaction mainly includes re-sintering of previous layers during scanning the newly added layer during the layer-by-layer fabrication. It is assumed that the sintering effect and the layer-interaction behavior can be transferred from a small patch to a long track as long as the size of the small patch is larger than the HAZ size. With the above assumption, we will train yNet based on dataset of small 128×128 patches ($>$ maximum HAZ depth of 117 pixels) and then demonstrate its spatial scalability to handling multiple layers of long tracks. Dataset of standard patches are obtained by cropping as-received simulation results every 10 pixels; see Fig. 6a. The detailed training and testing procedure can be found in Supplementary Note 1.

Testing on small standard patches

Fig. 6a illustrates the data generation of standard patches for training and testing from raw simulation results. The generated dataset is split into training and testing parts based on laser power and scanning speed sample points, i.e., 75 $[P, V]$ conditions randomly selected for training and the other 25 for testing; see detailed partition of $[P, V]$ points in Fig. S6. Fig. 6b shows 5 testing results on standard patches. yNet prediction results closely resemble physics-based simulations and only minor morphological error (see last row of Fig.6b) can be observed through pixel-wise comparison of their phase variables. In addition to visual resemblance, we examine the pixel-wise global accuracy (i.e., percentage of correct pixels) for such two-phase high-contrast fields. yNet achieves an as high as 99.13% similarity to ground truth in average for all 43500 testing results.

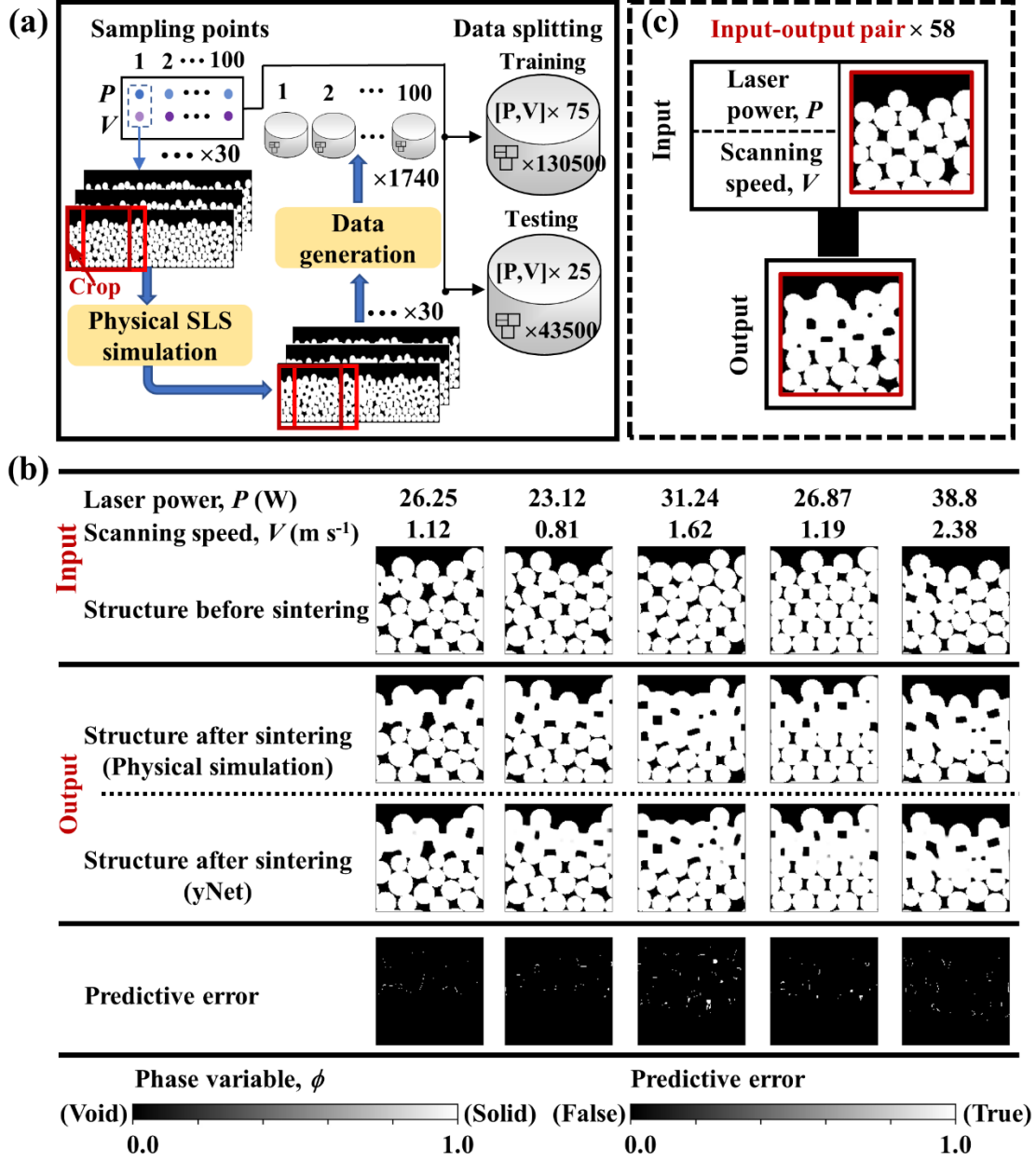


Fig.6 Testing of yNet on small standard patches. (a) pre-processing raw simulation results for dataset of standard patches. (b) testing results of yNet on standard patches. We randomly selected 5 out of 42750 testing results.

Testing on long tracks and spatial extrapolation

The as-trained yNet is scalable to perform large- or even extreme-scale simulation in space, such as multiple layers of long tracks. To validate step-by-step, we first examine

the spatial extrapolation of the trained yNet to handling a single-layer long track; see Fig. 7a. Fig. 7b compares the accuracy of yNet in modeling small patches of original size and long tracks. For different laser power conditions, yNet consistently shows a comparably high accuracy in dealing with long tracks. On average, yNet achieves a pixel-wise accuracy of 99.04% in modeling long tracks, which is close to 99.24% in modeling small patches. The root reason for its scalability to track of any length is that yNet essentially learns sintering effect associated with laser conditions and independent from track length. The second step is the further validation of sintering multi-layer long tracks that involves inter-layer interaction. Fig. S7 show an example of detailed simulation process for several layers. By adopting such layer-wise simulation, we obtain the multi-layer sintering results by yNet and physics-based model under different laser conditions; see Fig. 7c. By increasing laser power, we can observe that results by yNet and physics-based model both show a transition from layer-scale unsintered region (see top-left inset) to small scattered unsintered areas (see bottom-left inset) and, eventually, to a fully sintered state (see bottom-right inset). The unsintered regions are located in between layers, due to insufficient sintering depth under small laser powers. The quantitative results of measured unsintered regions (see top-right figure) further prove the good agreement between yNet-based and physical multi-layer simulations.

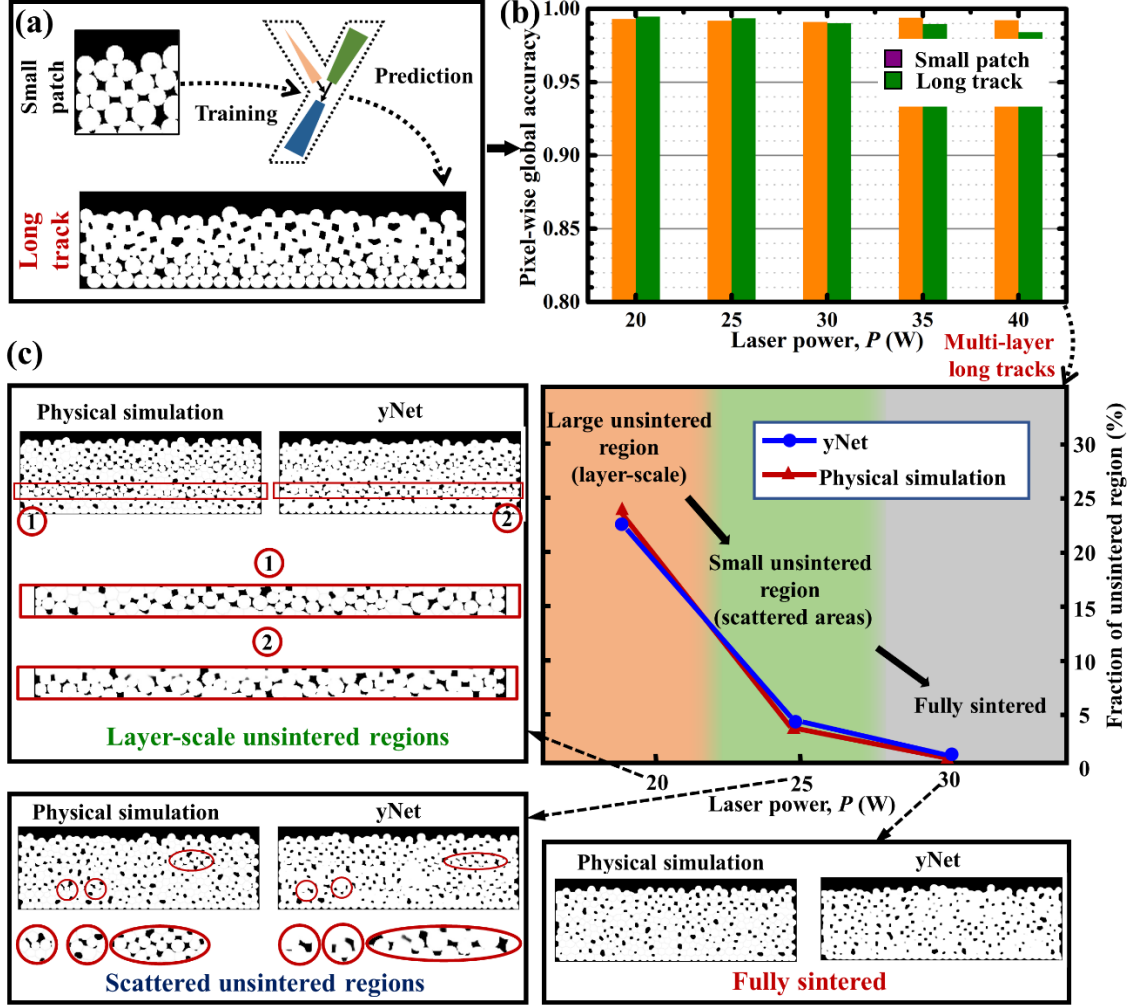


Fig. 7 Testing of yNet on extrapolating prediction for long tracks. (a) schematic illustration of as-trained yNet for predicting sintering of long track. (b) testing results of yNet for predicting sintering of long track under different laser conditions. (c) further extension of yNet to predicting sintering of multi-layer long tracks under different laser conditions. By following the workflow in Fig. S7, we perform sintering simulation of multiple layers under three laser power conditions ($P = 20, 25$, and 30 W). Other settings are the same as that used in Fig. S7. The overall sintering effect of multiple layers (measured by fraction of unsintered region) as a function of laser power are then obtained for yNet and physics-based model as shown. Note that, for unbiased measurement of unsintered regions, unsintered regions are manually labeled and calculated by the third party.

The above-validated spatial scalability can help address the computationally daunting task of full-component SLS simulation. As an illustration, we use yNet to

perform multi-layer sintering simulation of a practical 315-layer SLS component with dimensions of $70.8 \times 44.1 \text{ mm}^2$ (or 35416×22050 pixels). As depicted in Fig. S8, we completely emulate the practical SLS process to perform the full-component simulation layer-by-layer, i.e., generating a new layer of raw powder bed, CAD-model guided selective sintering simulation of the newly added layer, and repeat the above two steps until completing the whole component. It should be pointed out that computational cost of raw powder bed generation increases exponentially with track length. For efficiency, we simplify powder bed generation during full-component simulation, by depositing a continuous series of mini-powder-beds till completing deposition of the entire layer. This will numerically create large pores regularly observed in between adjunct mini-powder-beds, as shown later. Fig. 8a shows the final SLS simulation result. Due to its extremely large size, here we can only present a fraction of the detail of the whole simulation result; see Fig. 8b. The periodically observed large pores are attributed to numerical artifacts due to the aforementioned simplified powder-bed generation. Once available, a more efficient powder bed generation model can be integrated with our yNet for better component-level simulation. It is known that AM simulation is often limited to a single layer or even a representative volume element (RVE). A practical simulation of the entire layer-by-layer AM process is rarely achieved so far. Fig. 8a represents one of the first achievements of component-level AM simulation to date. Fig. 8c summarizes the simulation size of some of existing 2D sintering simulation research [51-55]. Although prior researches did not report computation time and hardware used, Fig. 8c gives a rough yet direct insight into the massively boosted simulation capability by yNet. The simulation size by yNet is several orders of magnitude larger than previous physics-based simulation. yNet enabled

component-level simulation would open up enormous possibilities in SLS practices. For example, SLS has been intentionally used for fabricating porous structures with high permeability and bio-compatibility [56, 57]. The trained yNet permits extensive virtual experiments in the pre-production phase, thus greatly facilitating structure design and process planning for fabricating a SLS component with any desired porosity distribution.

Comparison of yNet with physics-based model

As discussed early, physics-based SLS simulation especially at the component level is quite computationally demanding. It thus presents as an ideal case to demonstrate the computational superiority of yNet over physics-based model. In this subsection, we look into the computation time of yNet and the original physics-based SLS model. Again, yNet and physics-based model are intentionally tested on the laptop used in the benchmark study in Application 1.

Fig. 8d shows the computation time as a function of the length of modeled powder bed. We can find that simulation of small tracks like 640-pixel long tracks in Fig. 7a is just a matter of tens of milliseconds by using yNet. In striking contrast, the original physics-based model used in this study will take a few hours to complete the same task in the same computational setting. The test for physics-based model stops at the length of 2048 pixels, because it can be foreseen that by modeling even longer track, the computation time will increase rapidly to days. With such limited computational resources, yNet can however easily handle extremely large layers with a maximum length of ~21248 pixels. Modeling even this large layer remains to be nearly instant for yNet. In this regarding, the component-level SLS simulation (i.e., hundreds of large layers), which

is previously a formidable task even for high-end computing facilities, is now accomplishable in a few minutes with yNet by simply using a laptop.

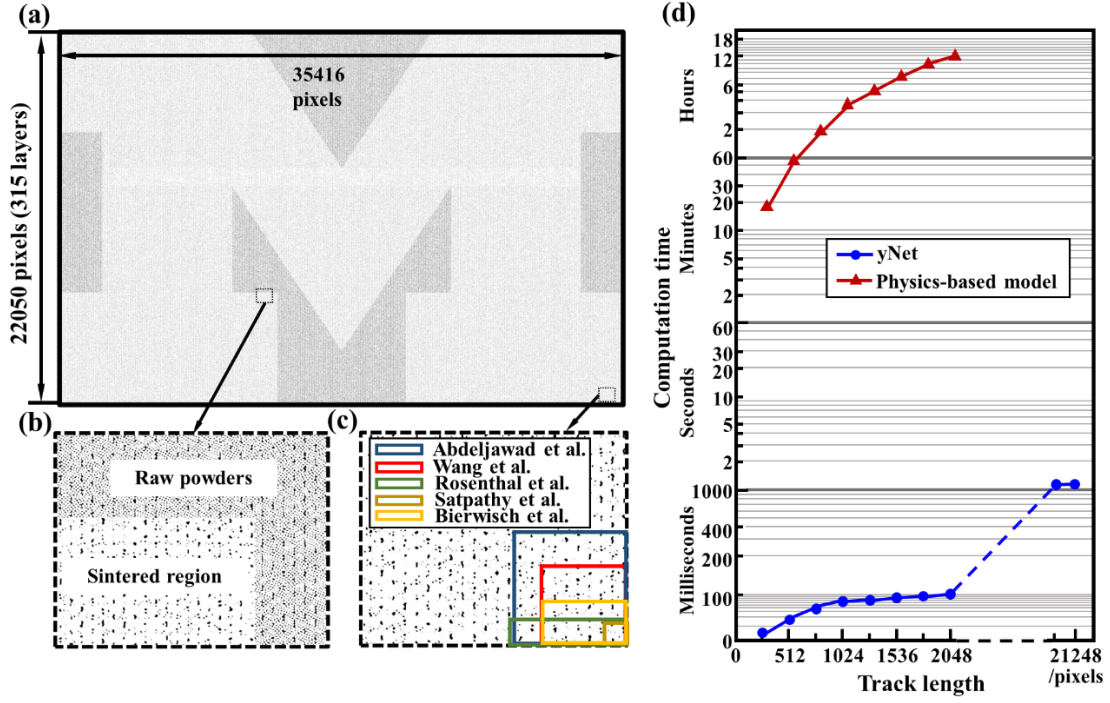


Fig. 8 yNet enabled component-level SLS simulation. (a) Final result of yNet based porosity simulation of a 315-layer high SLS component - block “M” logo of University of Michigan (35416×22050 pixels). (b) Magnified image showing the sintering details of a fraction of the SLS component. (c) Relative simulation size of existing sintering simulations to y-Net based component-level simulation. They are compared in terms of dimensionless size in number of pixels. (d) Comparison of computation time between yNet and physics-based model. In general, yNet is not only 5 to 6 orders of magnitude faster than physics-based model, but also able to handle extremely long tracks with even limited computational resource.

Application 3: stress field development

In this section, we quickly present another application of yNet to predicting stress field development. The main objective of this application is to test the robustness of yNet in (1) handling input and output field with striking difference and (2) extending the stress

field prediction for structures beyond the specific type of shape used in training. Stress field development is a totally different type of field evolvement problem compared to the previous two. The task in this application is to construct mapping between fields of different physical meaning - structural field and stress field. Compared to standard field evolvement of a certain type of field, the stress field development shows a radical difference between input and output images. The input structure is a simple high-contrast field, whereas the output stress field is characterized with complex texture and great variability. Therefore, this application may further push the performance limit of the proposed yNet in the tough application scenario of strong “field evolvement”.

Training and testing procedure

For illustration purpose, we trained yNet to predict stress field development within typical perforation structure (i.e., $0.1\text{m} \times 0.1\text{m}$ solid structure with a hole), when subject to compressive loading. Supplementary Note 2 describes the detailed training and testing procedure. As illustrated in Fig. 9a, the basic idea is to train yNet with a specific type of perforation structures, which all feature elliptic hole albeit with random combination of hole orientation, size and aspect ratio. Once trained, yNet is expected to predict stress field development for perforation structures with holes of not only elliptical shapes but also other geometries.

We first routinely test yNet for perforation structures with elliptical holes unused during training; see Fig. S11 for more details. Briefly, the prediction by yNet overall agrees well with FE-based simulation. The RMSE for all 1000 testing results is as small as 11.29 MPa. Therefore, although the input and output images are totally different and display huge changes, yNet can still correctly predict those significant “evolvements”.

Testing on other types of perforation structures (extrapolation???)

Besides the routine testing, we further test yNet for perforation structure with other categories of holes, including triangle, rectangle, and polygonal ones. For the 5 random testing results in Fig. 9b, the RMSE is 20.97, 50.43, 49.66, 41.33, 51.71 MPa respectively, which are relatively higher than that of testing with respect to elliptical holes. Although the predictive performance shows a small decrease, from the comparison of detailed stress distribution, yNet still reasonably predicts stress field development. It is thus believed that yNet has learned the basic rule for stress field development. For example, stress concentration tends to take place during sharp geometrical transitions along the loading direction and also initiates from two bottom corners with increasing loading (associated with fixed boundary at the bottom in the current modeling); the discontinuous geometrical changes transverse to the loading direction will however not serve as significant stress risers. Those rules are fully contained in the training dataset and universally valid for perforation structures of any type. Learning those fundamental rules thus endows yNet great extrapolation capability in coping with different perforation structures.

Noteworthy is that, such a data-driven stress field prediction capability also naturally addresses the data-driven prediction of stress hotspot. The prior art of this research area is grain-wise stress-hotspot prediction for grain microstructures [58, 59], which also relies on expert knowledge to hand-design contributing factors to stress hotspot. The yNet trained herein can predict the detailed distribution of loading-dependent stress field, thus allowing for direct pixel-level prediction of stress hotspot by simply thresholding using the critical stress.

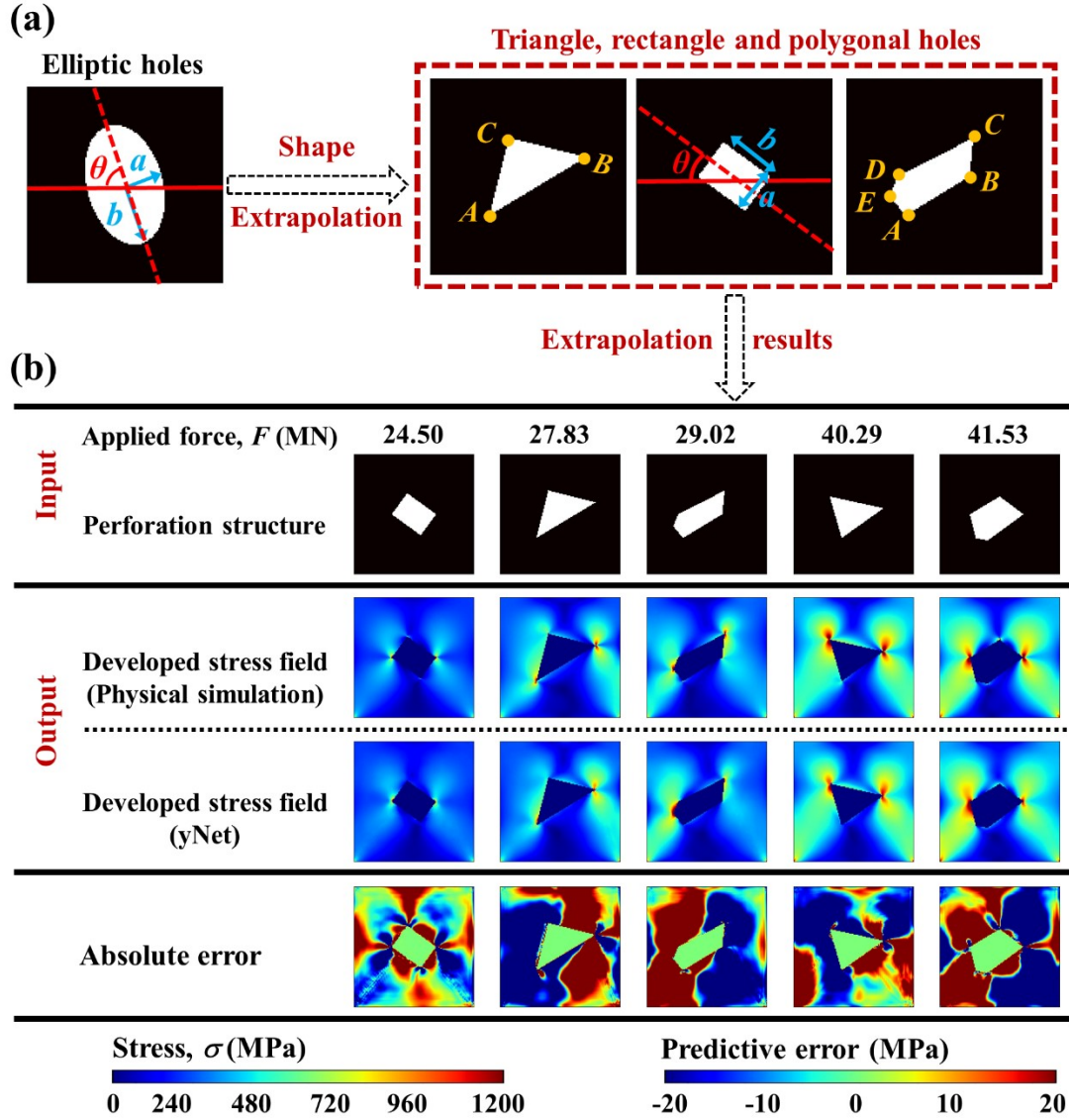


Fig.9 Illustrates the extrapolation capability in predicting stress development of various types of perforation structures. (a) yNet is trained based on simple dataset containing elliptic holes only. The as-trained yNet can predict stress field for other types of holes. All of the holes used in training and testing are randomly generated based on the descriptive methods as shown. (b) Testing of yNet on different types of perforation structures. In this second-round testing, these perforation structures have completely different types of holes than that in training dataset. Five randomly generated rectangle, triangle and polygonal holes are tested.

Discussion

Future applications

The proposed yNet should be generally applicable to a plenty of condition-dependent field evolvement and development problems. The presented applications have helped comprehensively examine the reliability of yNet. Those different application scenarios should cover many field evolvement and development problems across physical science and engineering. As the results throughout this study show, the proposed yNet is widely useful for simulating field evolvement in those different situations. The well-tested yNet thus has an enormous potential in future applications in a lot of domains. For example, Supplementary Note 3 describes another interesting application of yNet for grain growth simulation. It again utilizes spatiotemporal scalability of yNet to enable long-term, large-scale grain growth simulation (60K grains) based on a laptop. yNet can be also easily extended to 3D modeling with simple modifications like adoption of 3D convolutions and 3D pooling operations. It is known that computational cost of physical simulation usually increases sharply for 3D, in which case yNet may become even more invaluable.

Limitations

The main limitation of yNet is that the trained yNet is sometimes limited in scaling to field of large and variable dimensions. In yNet-based component-level SLS simulation, the scalability to long tracks relies on the assumption of globally uniform evolving kinetics of porosity evolution in SLS. That is, the sintering effect depends on applied laser condition and is consistent along the length of the sintered track for a given

laser condition; see Fig. S9. The large-scale simulation of natural grain growth is also based on its completely uniform evolving kinetics over modeling domain. This is however not always the case, e.g., fluid dynamics simulation and stress field prediction where such simple scalability may not hold. Other solutions are demanded for developing more advanced scientific ConvNets. However, the core of yNet is still expected to play a role whenever it comes to developing multi-input ConvNets and merging input signals.

Another limitation is that only parameter-type conditions are considered and demonstrated in the current study. There are however higher-dimensional and more complex conditions in various physical and engineering problems. For example, the applied loading in stress-hotspot development case in Fig. 8a can be non-uniform, thus not describable by a single scalar parameter. Also, besides natural grain growth in Fig. 8b, grain growth in practice can be influenced by their living temperature field, as the preferred growth direction is along the local thermal gradient [60]. Nonetheless, those complex conditions, no matter how many dimensions they have, can be eventually converted to an embedding vector of desired length. It would then allow for the adoption of gating-based signal merging for efficient incorporation of their impacts on field evolution.

Conclusion

In summary, we have proposed a multi-input deep convolutional network, yNet, which aims to solidly push our data-driven modeling capability to multi-input field-to-field/image-to-image regression. It provides an ultra-fast way to realistically simulate various condition-dependent field evolvments, by treating them as a purely data-driven

regression problem. Experiments demonstrate that yNet enjoys extreme computational efficiency against the physics-based model and significant computational advantage over its ConvNet counterparts. The proposed yNet is applied to three representative data-driven modeling problems - fluid dynamics, porosity evolution in selective laser sintering (SLS), and stress field development. Those diverse applications have demonstrated the general effectiveness of yNet in handling many types of field evolvments in physical science and engineering. Therefore, the well-tested yNet may have a profound impact as a conceptually simple, light-weight yet generally useful multi-input field-to-field regression tool.

Acknowledgements

The research is financially supported by XX. Start-up and NSF (bi-continuous see wenhua's paper)

Author contribution

Z.W. designed yNet, conducted yNet based simulations, prepared figures and performed analysis. L.Y. X and Z.O. conducted extensive fluid flow simulations and provided relevant dataset. W.H.Y. and Y.C.L. carried out extensive physics-based SLS simulations and preparation of relevant datasets. X.W. help with developing the multi-physical porosity simulation model. Y.J.Z. and P.W.L performed extensive grain growth simulations and prepared relevant dataset. Z.W. performed massive stress-field development simulations for generating relevant dataset. Y.H.X. contributed to powder bed generation in full-component porosity simulation. L.C. conceived the application of

convolutional network to simulating field evolvments and supervised this research. Z.W., L. Y. X., W.H.Y. and X.W. wrote the paper. Z.W. prepared the supplementary notes. Y.J.Z. and P.W. L. prepared Supplementary Video. All authors discussed results and commented on the manuscript.

Method

Physics-based fluid dynamics model

XX

Physics-based SLS model

Physics-based SLS simulation provides training and testing data. In this study, we basically customize a phase-field-based sintering model [49] for applicability in SLS, with further incorporation of heat transfer model and a Gaussian heat source model describing the effective heat input from moving laser beam [61]. The sintering model itself has properly taken into account multiple physical processes, by first reformulating the effective diffusion coefficient as:

$$D^{eff} = D_{vol}\phi(\rho) + D_{vap}[1 - \phi(\rho)] + D_{sf}\rho(1 - \rho) + D_{gb}\sum_{\alpha}\sum_{\alpha' \neq \alpha}(\eta_{\alpha}\eta_{\alpha'}) \quad (3)$$

where, D_{vol} , D_{vap} , D_{sf} and D_{gb} are temperature-dependent diffusion coefficients in solid volume, vapor, along surface, and grain boundary, respectively. η_{α} is phase variable that describes the α -th particle. In addition, the non-conserved porosity evolution is partially governed by the rigid body motion-induced advection velocity field, which is a combination of the translation of rotation from all relevant particles,

$\mathbf{v}_{rbm} = \sum_{\alpha} [\mathbf{v}_t(\alpha) + \mathbf{v}_r(\alpha)]$. The velocity fields of rigid-body translation, $\mathbf{v}_t$ and rotation, $\mathbf{v}_r$,

of the α -th particle are calculated by:

$$\mathbf{v}_t(\mathbf{r}, \alpha) = \frac{m_t}{V(\alpha)} \mathbf{F}(\alpha) \eta_{\alpha}(\mathbf{r}) \quad (4)$$

$$\mathbf{v}_r(\mathbf{r}, \alpha) = \frac{m_r}{V(\alpha)} \mathbf{T}(\alpha) [\mathbf{r} - \mathbf{r}_c(\alpha)] \eta_{\alpha}(\mathbf{r}) \quad (5)$$

where $m_t = 500$ and $m_r = 1$. The force, $\mathbf{F}$, and torque, $\mathbf{T}$, acting on the powder particles are associated with their real-time structural environment.

The operating range of laser power is [25, 40] W and scanning speed [0.5, 2.5] m s⁻¹. The alloy we used is stainless steel 316L. The spatial and temporal simulation resolution are $\Delta x = \Delta y = 2\mu\text{m}$ and $\Delta t = 1\mu\text{s}$, respectively. Full mathematical details of our physical SLS model are reported in a separate paper on multi-physical SLS simulation [50].

Powder generation model

In addition to sintering model for simulating porosity evolution, generation of powder bed (i.e. initial porous structure) is simulated using a “rain” model [62]. The basic mechanism of rain model is to add a powder particle to the lowest position on the surface of the current powder layer, update the powder layer and repeat the above two steps. This iterative process is terminated when the newly added particle reaches the user-defined layer height. The mean and standard deviation of the diameter of deposited powders are $25\mu\text{m}$ and $0.5\mu\text{m}$, respectively. It should be pointed out that yNet in this study will be trained to simulate porosity evolution and thus replace physics-based

sintering model only. Powder bed generation model is a separate model that provides initial structure for physical sintering simulation and yNet-based sintering simulation.

yNet

We use the yNet instantiated in fluid dynamics simulation as an illustrative example. The detailed architecture is shown in Fig. S1a. The first layer of MLP is the condition input (i.e, Reynolds number), the hidden layer is a fully connected layer with 256 neurons and the final layer 512 neurons. For the encoder, the input is vorticity flow field at the current step in 256×64 image. Each green block represents a combo of Conv+Relu operations and light green block means max-pooling. In this manner, encoder finally yields 512×2 feature maps, which are passed to a dropout layer with rate of 0.5 before merging with MLP signal. The decoder just has a somewhat mirrored topology of encoder, with each blue block representing a combo of Conv+Relu operations and dark blue block up-sampling. For the other three applications, we basically use the above-described yNet, but with potentially adjusted depth of encoder-decoder and width of MLP depending on the modeling complexity.

Two types of loss functions are used in this study. For output field with strong texture (i.e, flow field and stress field), we use mean squared error (MSE) to evaluate loss:

$$Loss = \frac{1}{N} \sum_{i=1}^N \left| f(\mathbf{a}^i, \mathbf{X}^i) - \bar{\mathbf{Y}}^i \right|^2 \quad (6)$$

where N is number of samples processed, $\mathbf{a}$ is the condition-parameter inputs, $\mathbf{X}$ is the field input, $f(\mathbf{a}^i, \mathbf{X}^i)$ is the evolved field by prediction and $\bar{\mathbf{Y}}$ is evolved field of ground

truth. For output field that clearly displays two distinct phases (i.e., porosity structure and grain structure), we use binary cross-entropy loss to better penalize predictive errors:

$$Loss = -\frac{1}{N} \sum_{i=1}^N \left| f(\mathbf{a}^i, \mathbf{X}^i) \cdot \log \bar{Y}^i + (1 - f(\mathbf{a}^i, \mathbf{X}^i)) \cdot (1 - \log \bar{Y}^i) \right|^2 \quad (7)$$

Reference

- [1] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *nature*, vol. 521, pp. 436-444, 2015.
- [2] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, *et al.*, "Handwritten digit recognition with a back-propagation network," in *Advances in neural information processing systems*, 1990, pp. 396-404.
- [3] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural computation*, vol. 29, pp. 2352-2449, 2017.
- [4] G. Du, K. Wang and S. Lian, "Vision-based robotic grasping from object localization, pose estimation, grasp detection to motion planning: A review," *arXiv preprint arXiv:1905.06658*, 2019.
- [5] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez, "A review on deep learning techniques applied to semantic segmentation," *arXiv preprint arXiv:1704.06857*, 2017.
- [6] Y. Guo, Y. Liu, T. Georgiou and M. S. Lew, "A review of semantic segmentation using deep neural networks," *International Journal of Multimedia Information Retrieval*, vol. 7, pp. 87-93, 2018/06/01 2018.
- [7] J. Ling, M. Hutchinson, E. Antono, B. DeCost, E. A. Holm, *et al.*, "Building data-driven models with microstructural images: Generalization and interpretability," *Materials Discovery*, vol. 10, pp. 19-28, 2017.
- [8] B. L. DeCost and E. A. Holm, "A computer vision approach for automated analysis and classification of microstructural image data," *Computational materials science*, vol. 110, pp. 126-133, 2015.
- [9] X. Li, Y. Zhang, H. Zhao, C. Burkhart, L. C. Brinson, *et al.*, "A transfer learning approach for microstructure reconstruction and structure-property predictions," *Scientific reports*, vol. 8, 2018.
- [10] R. Cang, Y. Xu, S. Chen, Y. Liu, Y. Jiao, *et al.*, "Microstructure representation and reconstruction of heterogeneous materials via deep belief network for computational material design," *Journal of Mechanical Design*, vol. 139, 2017.
- [11] W. Yang, Z. Wang, T. Yang, L. He, X. Song, *et al.*, "Exploration of the Underlying Space in Microscopic Images via Deep Learning for Additively Manufactured Piezoceramics," *ACS Applied Materials & Interfaces*, 2021/09/01 2021.
- [12] F. Imani, R. Chen, E. Diewald, E. Reutzel, and H. Yang, "Deep learning of variant geometry in layerwise imaging profiles for additive manufacturing quality control," *Journal of Manufacturing Science and Engineering*, vol. 141, 2019.
- [13] Q. Tian, S. Guo, E. Melder, L. Bian, and W. Guo, "Deep Learning-Based Data Fusion Method for In Situ Porosity Detection in Laser-Based Additive Manufacturing," *Journal of Manufacturing Science and Engineering*, vol. 143, 2021.

- [14] C. Herriott and A. D. Spear, "Predicting microstructure-dependent mechanical properties in additively manufactured metals with machine-and deep-learning methods," *Computational Materials Science*, vol. 175, p. 109599, 2020.
- [15] Z. Yang, Y. C. Yabansu, R. Al-Bahrani, W.-k. Liao, A. N. Choudhary, *et al.*, "Deep learning approaches for mining structure-property linkages in high contrast composites from simulation datasets," *Computational Materials Science*, vol. 151, pp. 278-287, 2018.
- [16] T. Le, V. C. Epa, F. R. Burden and D. A. Winkler, "Quantitative structure–property relationship modeling of diverse materials properties," *Chemical Reviews*, vol. 112, pp. 2889-2919, 2012.
- [17] W. Yan, S. Lin, O. L. Kafka, C. Yu, Z. Liu, *et al.*, "Modeling process-structure-property relationships for additive manufacturing," *Frontiers of Mechanical Engineering*, vol. 13, pp. 482-492, December 01 2018.
- [18] K. Hasegawa, K. Fukami, T. Murata and K. Fukagata, "CNN-LSTM based reduced order modeling of two-dimensional unsteady flows around a circular cylinder at different Reynolds numbers," *Fluid Dynamics Research*, vol. 52, p. 065501, 2020.
- [19] K. Yang, Y. Cao, Y. Zhang, S. Fan, M. Tang, *et al.*, "Self-supervised learning and prediction of microstructure evolution with convolutional recurrent neural networks," *Patterns*, vol. 2, p. 100243, 2021.
- [20] A. Frankel, K. Tachida and R. Jones, "Prediction of the evolution of the stress field of polycrystals undergoing elastic-plastic deformation with a hybrid neural network model," *Machine Learning: Science and Technology*, vol. 1, p. 035005, 2020.
- [21] M. Raj, S. Thakre, R. K. Annabattula and A. K. Kanjarla, "Estimation of Local Strain Fields in Two-Phase Elastic Composite Materials Using UNet-Based Deep Learning," *Integrating Materials and Manufacturing Innovation*, pp. 1-17, 2021.
- [22] Y. Zhu and N. Zabaras, "Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification," *Journal of Computational Physics*, vol. 366, pp. 415-447, 2018/08/01/ 2018.
- [23] Y. He, F. Fei, W. Wang, X. Song, Z. Sun, *et al.*, "Predicting manufactured shapes of a projection micro-stereolithography process via convolutional encoder-decoder networks," in *ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2018, pp. V01BT02A033-V01BT02A033.
- [24] Z. Shen, X. Shang, M. Zhao, X. Dong, G. Xiong, *et al.*, "A learning-based framework for error compensation in 3D printing," *IEEE transactions on cybernetics*, vol. 49, pp. 4042-4050, 2019.
- [25] X. Guo, W. Li and F. Iorio, "Convolutional neural networks for steady flow approximation," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 481-490.
- [26] J. R. Mianroodi, N. H. Siboni and D. Raabe, "Teaching solid mechanics to artificial intelligence—a fast solver for heterogeneous materials," *npj Computational Materials*, vol. 7, pp. 1-10, 2021.
- [27] B. P. Croom, M. Berkson, R. K. Mueller, M. Presley, and S. Storck, "Deep learning prediction of stress fields in additively manufactured metals with intricate defect networks," *arXiv preprint arXiv:2105.10564*, 2021.
- [28] O. Ronneberger, P. Fischer and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, 2015, pp. 234-241.
- [29] J. Long, E. Shelhamer and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431-3440.
- [30] Z. Yuan, Y. Jiang, J. Li and H. Huang, "Hybrid-DNNs: Hybrid Deep Neural Networks for Mixed Inputs," *arXiv preprint arXiv:2005.08419*, 2020.

- [31] S. Law, B. Paige and C. Russell, "Take a look around: using street view and satellite images to estimate house prices," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, pp. 1-19, 2019.
- [32] F. Allaire, V. Mallet and J.-B. Filippi, "Emulation of wildland fire spread simulation using deep learning," *Neural networks*, vol. 141, pp. 184-198, 2021.
- [33] L. A. Hendricks, S. Venugopalan, M. Rohrbach, R. Mooney, K. Saenko, *et al.*, "Deep compositional captioning: Describing novel object categories without paired training data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1-10.
- [34] V. Badrinarayanan, A. Kendall and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, pp. 2481-2495, 2017.
- [35] H. Noh, S. Hong and B. Han, "Learning deconvolution network for semantic segmentation," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1520-1528.
- [36] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *arXiv preprint arXiv:1412.7062*, 2014.
- [37] Z. Nie, H. Jiang and L. B. Kara, "Stress Field Prediction in Cantilevered Structures Using Convolutional Neural Networks," *Journal of Computing and Information Science in Engineering*, vol. 20, 2019.
- [38] S. Bhatnagar, Y. Afshar, S. Pan, K. Duraisamy, and S. Kaushik, "Prediction of aerodynamic flow fields using convolutional neural networks," *Computational Mechanics*, vol. 64, pp. 525-545, 2019.
- [39] J. Hu, L. Shen and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132-7141.
- [40] K. Hasegawa, K. Fukami, T. Murata and K. Fukagata, "Machine-learning-based reduced-order modeling for unsteady flows around bluff bodies of various shapes," *Theoretical and Computational Fluid Dynamics*, vol. 34, pp. 367-383, 2020.
- [41] S. Lee and D. You, "Prediction of laminar vortex shedding over a cylinder using deep learning," *arXiv preprint arXiv:1712.07854*, 2017.
- [42] P. Baldi and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima," *Neural networks*, vol. 2, pp. 53-58, 1989.
- [43] S. L. Brunton, B. R. Noack and P. Koumoutsakos, "Machine learning for fluid mechanics," *Annual Review of Fluid Mechanics*, vol. 52, pp. 477-508, 2020.
- [44] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, 2014, pp. 818-833.
- [45] T. Murata, K. Fukami and K. Fukagata, "Nonlinear mode decomposition with convolutional neural networks for fluid dynamics," *Journal of Fluid Mechanics*, vol. 882, 2020.
- [46] K. Fukami, T. Nakamura and K. Fukagata, "Convolutional neural network based hierarchical autoencoder for nonlinear mode decomposition of fluid field data," *Physics of Fluids*, vol. 32, p. 095110, 2020.
- [47] S. Kumar, "Selective laser sintering: A qualitative and objective approach," *JOM*, vol. 55, pp. 43-47, 2003/10/01 2003.
- [48] H. Dürr, R. Pilz and N. S. Eleser, "Rapid tooling of EDM electrodes by means of selective laser sintering," *Computers in Industry*, vol. 39, pp. 35-45, 1999.
- [49] Y. U. Wang, "Computer modeling and simulation of solid-state sintering: A phase field approach," *Acta materialia*, vol. 54, pp. 953-961, 2006.

- [50] X. Wang, Y. Liu, L. Li, C. O. Yenusah, Y. Xiao, *et al.*, "Multi-scale phase-field modeling of layer-by-layer powder compact densification during solid-state direct metal laser sintering," *Materials & Design*, vol. 203, p. 109615, 2021.
- [51] F. Abdeljawad, D. S. Bolintineanu, A. Cook, H. Brown-Shaklee, C. DiAntonio, *et al.*, "Sintering processes in direct ink write additive manufacturing: A mesoscopic modeling approach," *Acta Materialia*, vol. 169, pp. 60-75, 2019.
- [52] Y. Wang, Y. Liu, C. Ciobanu and B. R. Patton, "Simulating microstructural evolution and electrical transport in ceramic gas sensors," *Journal of the American Ceramic Society*, vol. 83, pp. 2219-2226, 2000.
- [53] W. S. Rosenthal, F. C. Grogan, Y. Li, E. I. Barker, J. F. Christ, *et al.*, "" Sintering" Models and In-Situ Experiments: Data Assimilation for Microstructure Prediction in SLS Additive Manufacturing of Nylon Components," *MRS Advances*, pp. 1-9.
- [54] B. B. Satpathy, J. Nandy and S. Sahoo, "Investigation of consolidation kinetics and microstructure evolution of Al alloys in direct metal laser sintering using phase field simulation," *IOP ConfSer Mater ScEng*, vol. 338, pp. 1-6, 2018.
- [55] C. Bierwisch, S. Mohseni-Mofidi, B. Dietemann, T. Kraft, J. Rudloff, *et al.*, "Particle-based simulation, dimensional analysis and experimental validation of laser absorption and thermo-viscous flow during sintering of polymers," *Procedia CIRP*, vol. 94, pp. 74-79, 2020.
- [56] T. Furumoto, A. Koizumi, M. R. Alkahari, R. Anayama, A. Hosokawa, *et al.*, "Permeability and strength of a porous metal structure fabricated by additive manufacturing," *Journal of Materials Processing Technology*, vol. 219, pp. 10-16, 2015/05/01/ 2015.
- [57] A. Basalah, Y. Shanjani, S. Esmaceli and E. Toyserkani, "Characterizations of additive manufactured porous titanium implants," *Journal of Biomedical Materials Research Part B: Applied Biomaterials*, vol. 100, pp. 1970-1979, 2012.
- [58] A. Mangal and E. A. Holm, "Applied machine learning to predict stress hotspots I: Face centered cubic materials," *International Journal of Plasticity*, vol. 111, pp. 122-134, 2018.
- [59] A. Mangal and E. A. Holm, "Applied machine learning to predict stress hotspots II: Hexagonal close packed materials," *International Journal of Plasticity*, vol. 114, pp. 1-14, 2019.
- [60] P. Liu, Y. Ji, Z. Wang, C. Qiu, A. Antonysamy, *et al.*, "Investigation on evolution mechanisms of site-specific grain structures during metal additive manufacturing," *Journal of Materials Processing Technology*, vol. 257, pp. 191-202, 2018.
- [61] Y. Yang, P. Kühn, M. Yi, H. Egger, and B.-X. Xu, "Non-isothermal Phase-Field Modeling of Heat–Melt–Microstructure-Coupled Processes During Powder Bed Fusion," *JOM*, 2020/01/17 2020.
- [62] L.-X. Lu, N. Sridhar and Y.-W. Zhang, "Phase field simulation of powder bed-based additive manufacturing," *Acta Materialia*, vol. 144, pp. 801-809, 2018.

