Patterns

# Multi-input convolutional network for ultrafast simulation of field evolvement

## Highlights

- A multi-input convolutional network is proposed to model evolution of physical fields
- The lightweight model shows general effectiveness in four diverse applications
- It also displays good extrapolative prediction beyond training datasets
- Full-component selective laser sintering and large grain growth modeling are given


## Authors

Zhuo Wang, Wenhua Yang, Linyan Xiang, ..., Mihaela Banu, Oleg Zikanov, Lei Chen

## Correspondence

leichn@umich.edu



## In brief

Wang et al. propose a conceptually simple multi-input convolutional network, yNet, as a general-purpose, multi-input, image-to-image regression tool that is widely useful in physical sciences and engineering. It is applied for ultrafast data-driven modeling of fluid dynamics, porosity evolution in selective laser sintering, stress field development within perforation structures, and grain growth. It shows general effectiveness in modeling condition-dependent evolution and development of physical fields. Large- and extreme-scale simulations that are often computationally impossible using physics-based models are also performed.


Wang et al., 2022, Patterns *3*, 100494
June 10, 2022 
https://doi.org/10.1016/j.patter.2022.100494

Patterns

CellPress
OPEN ACCESS

Article

# Multi-input convolutional network for ultrafast simulation of field evolvement

Zhuo Wang,[1,7] Wenhua Yang,[1,2,7] Linyan Xiang,[1] Xiao Wang,[3] Yingjie Zhao,[4] Yaohong Xiao,[1] Pengwei Liu,[4] Yucheng Liu,[2,5] Mihaela Banu,[6] Oleg Zikanov,[1] and Lei Chen[1,8,*]

[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
[5]Department of Mechanical Engineering, South Dakota State University, Brookings, SD 57007, USA
[6]Department of Mechanical Engineering, University of Michigan, Ann Arbor, MI 48128, USA
[7]These authors contributed equally
[8]Lead contact
*Correspondence: leichn@umich.edu
https://doi.org/10.1016/j.patter.2022.100494

**THE BIGGER PICTURE** In physical sciences and engineering, the convolutional network (ConvNet) has been used increasingly to simulate the evolvement of physical fields, e.g., flow field evolvement. Physical field data are fed as images, and ConvNet treats the field evolvement as a field-to-field/image-to-image regression problem, i.e., building the mapping from the input flow field to the evolved flow field. The ConvNet, when trained, can be a cheap substitute for physics-based models, enabling fast simulation of field evolvement. However, a big challenge still lies in incorporating conditions that dictate field evolvement, e.g., fluid properties associated with fluid dynamics. We propose a light multi-input ConvNet as a general-purpose, multi-input, image-to-image regression tool. Its simplicity and usefulness are demonstrated by modeling various condition-dependent field evolvements and developments. Large- and extreme-scale simulations are also performed based on its computational superiority.


**Development/Pre-production:** Data science output has been rolled out/validated across multiple domains/problems

## SUMMARY

There is a compelling need for the regression capability of 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 capability can provide a maximum to full substitute of a physics-based model, enabling fast simulation of various field evolvements. We propose a conceptually simple, lightweight, but powerful multi-input convolutional network (ConvNet), yNet, that merges multi-input signals by manipulating high-level encodings of field/image input. yNet can significantly reduce the model size compared with its ConvNet counterpart (e.g., to only one-tenth for main architecture of 38-layer depth) and is as much as six orders of magnitude faster than a physics-based model. yNet is applied for data-driven modeling of fluid dynamics, porosity evolution in sintering, stress field development, and grain growth. It consistently shows great extrapolative prediction beyond training datasets in terms of temporal ranges, spatial domains, and geometrical shapes.

## INTRODUCTION

A deep convolutional network (ConvNet) utilizes stacks of convolutional layers for automatic, hierarchical representation learning of high-dimensional data. The convolutional components of ConvNet bring great advantages when 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 remarkable application in digit recognition nearly 30 years ago,[2] ConvNet has served as the backbone for performing various computer vision (CV) and image-centered tasks.[3–6] Upon realization of its distinct advantage in image modeling, there has been a surge of applications in different scientific domains where image-involved problems

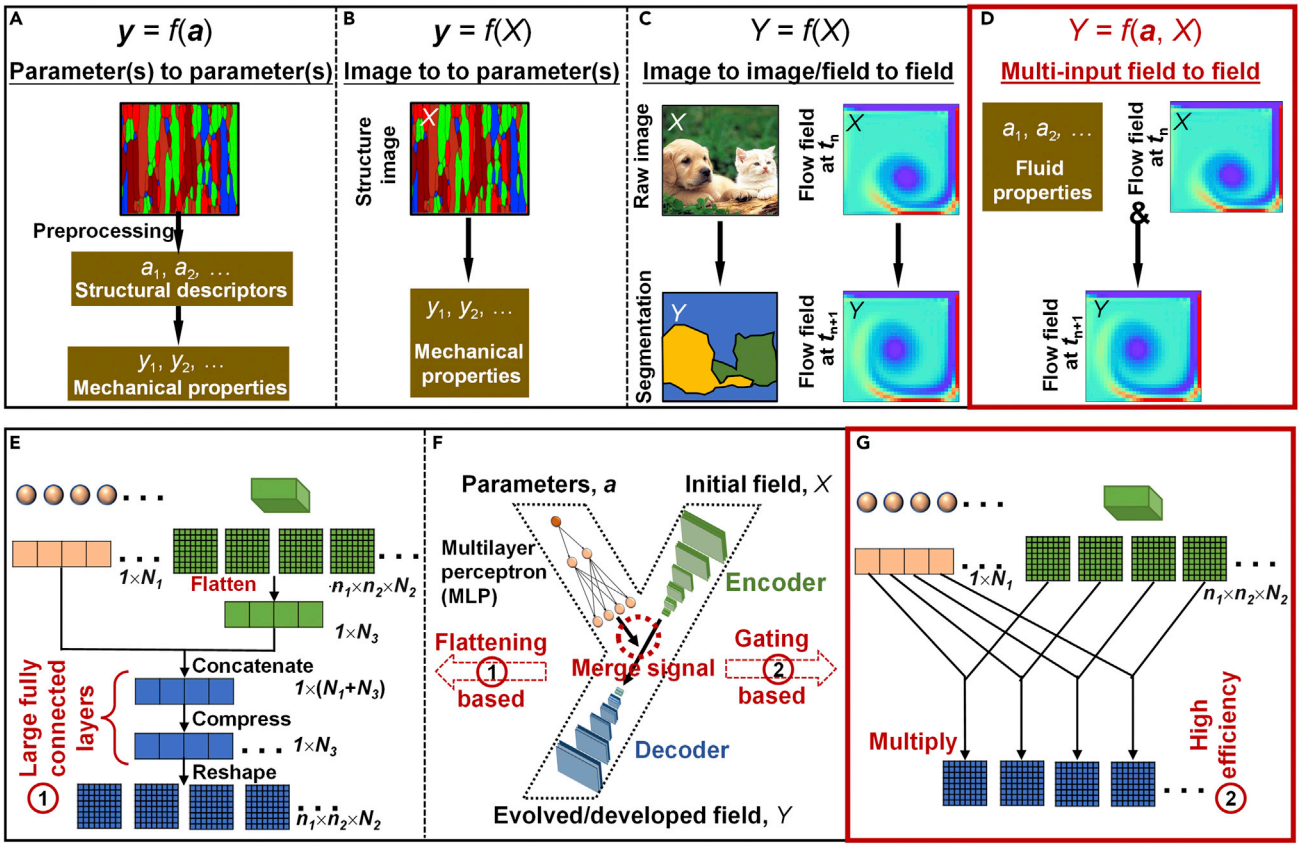


**Figure 1. Four different regression tasks with increasing complexity and multi-input ConvNet**

(A) Parameter(s)-to-parameter(s) regression for conventional structure-property modeling, where 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 an image and output is parameters.

(C) Image-to-image/field-to-field regression for semantic segmentation and fluid dynamics simulation, where the input and output are images.

(D) Multi-input field-to-field regression for condition-incorporated fluid dynamics simulation, where the mixed input is parameters of fluid properties and an image of the current flow field and the output is an image of the next-step flow field.

(E–G) The proposed y-shaped multi-input ConvNet for handling the multi-input field-to-field regression tasks (F) and comparison of the conventional flattening-based (E) and the proposed gating-based (G) signal merging strategy.

are ubiquitous. Some of these scientific applications include image classification with respect to scientific images,[7,8] microstructure characterization and reconstruction (MCR),[9–11] and 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 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] whereas ConvNets directly take raw structure information as input and can construct a high-fidelity relationship between image-based structure and property (Figures 1A and 1B).

Another prominent application is ConvNet-based simulation of field evolvement and development. This also takes advantage of its superior image modeling capability because different types of physical fields, such as temperature, velocity, density fields, and structures described by various physical variables, can eventually be presented as images. In this regard, ConvNets treat field evolvement and development as a purely data-driven, image-to-image/field-to-field regression problem (Figure 1C). Some typical examples include prediction of transient fluid dynamics (i.e., current flow field to future flow field[18]), microstructure evolution (i.e., current structure to future structure[19]), and stress/strain field development (structure to stress/strain field[20,21]). ConvNets simply aim to build these field-to-field mappings from training datasets without using any physical differential equations. Therefore, the ConvNet, when trained, can be an inexpensive substitute for a physics-based model, enabling fast but realistic simulations of field evolvement 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 used for semantic segmentation, is inadequate when it comes to modeling field evolvements. In semantic segmentation, the output image or segmentation result is uniquely determined by the input image to be segmented (Figure 1C). However, the evolvement and development of a physical field are usually not only decided by the initial state but also controlled by the applied conditions. For instance, a flow

field developed at a future point is concerned with the current flow field as well as fluid properties (Figure 1D), and stress field development greatly depends on the condition of applied force. With a pure image-to-image regression tool, those existing research have to fix conditions, thus failing to incorporate condition-related parameters. 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 prevents wide use of the trained ConvNet. Developing a more general multi-input field-to-field regressor that can build a functional relationship between multi-input (i.e., input field and condition parameters) and output fields would be of huge significance. It will, for example, allow us to train one single ConvNet applicable to flow fields with distinct conditions (Figure 1D) instead of training separate ConvNets for each. 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 shown in Figures 1C and 1D.

We propose a multi-input deep ConvNet that naturally exhibits a signature Y shape as a multi-input field-to-field regressor (Figure 1F). We thus call it yNet, which conveys its most salient feature, efficient fusion of an additional input signal of condition-related parameters, in comparison with the pure image-to-image regression neural network (I-shaped) in the aforementioned CV and scientific applications (Figure 1C). The main architecture is basically composed of an encoder-decoder based on a deep ConvNet and a branch of multi-layer perceptron (MLP). The encoder essentially plays the 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 (Figure 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 described later. The decoder serves to correctly reconstruct the merged signal back to a meaningful and desired output field through the deep deconvolution process. 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 induced by coarsening during max-pooling in the encoding process and improve information flow from (image) input to decoder.

After building the main architecture, merging signals of multiple inputs poses a key problem. In fact, such multiple mixed inputs (a high-dimensional image and scalar parameters) are not uncommon in various machine learning (ML) tasks, although the output is not necessarily an image. However, the merging strategies in existing multi-input ConvNets can suffer from various drawbacks associated with dense parameters. A common merging strategy[30–32] is to flatten the image signal, here represented by its feature maps, into a long vector, making it compatible and, hence, concatenable to the vectorial output of MLP (Figure 1E). The flattening-based strategy is suitable and 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, that are known to be parameter intensive. The dense parameters can bring about drawbacks, including increased training complexity requiring sophisticated training schemes, more hardware resources demanded during training and inference, and slow inference speed. This is clearly shown, for example, in comparative studies[33] between the light-weight SegNet and other semantic segmentation ConvNets.[29,34,35] Apart from the parameter-efficiency issue, the flattening operation will interrupt the signal flow of 2D feature maps, which themselves are meaningful as spatial features in 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 steps to force the long vector back to 2D useful maps for the decoding process. A flattening-based strategy, if workable for multi-input image-to-image regression,[36,37] might make the multi-input ConvNet more like a brute-force regressor.

To overcome the shortcomings of the flattening-based merging strategy, we propose to properly merge signals using a one-to-one connection via multiplication (Figure 1G), which is mathematically akin to the excitation/gating mechanism in SENet for improving image classification.[38] The proposed gating-based signal merging strategy features stunning 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 toward high-level representation of the output field. The effectiveness of the gating-based merging strategy lies in the fact that these neurons interact with respective feature maps in a direct, neat, and therefore strong manner. 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 a given condition input; the decoder then appropriately reconstructs the gated feature maps into the realistic evolved or developed field.

This research focuses on 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 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," falling within the interest of the proposed yNet.

Next, we present different applications of yNet in physical sciences and engineering, which collectively demonstrate the general effectiveness of yNet in handling many field evolvement and development problems. Following these diverse applications, we finally discuss its enormous potential for future applications as well as some limitations.

## RESULTS

### Fluid dynamics

We start with data-driven modeling of flow fields over a cylinder using the proposed yNet. This fluid dynamics problem is

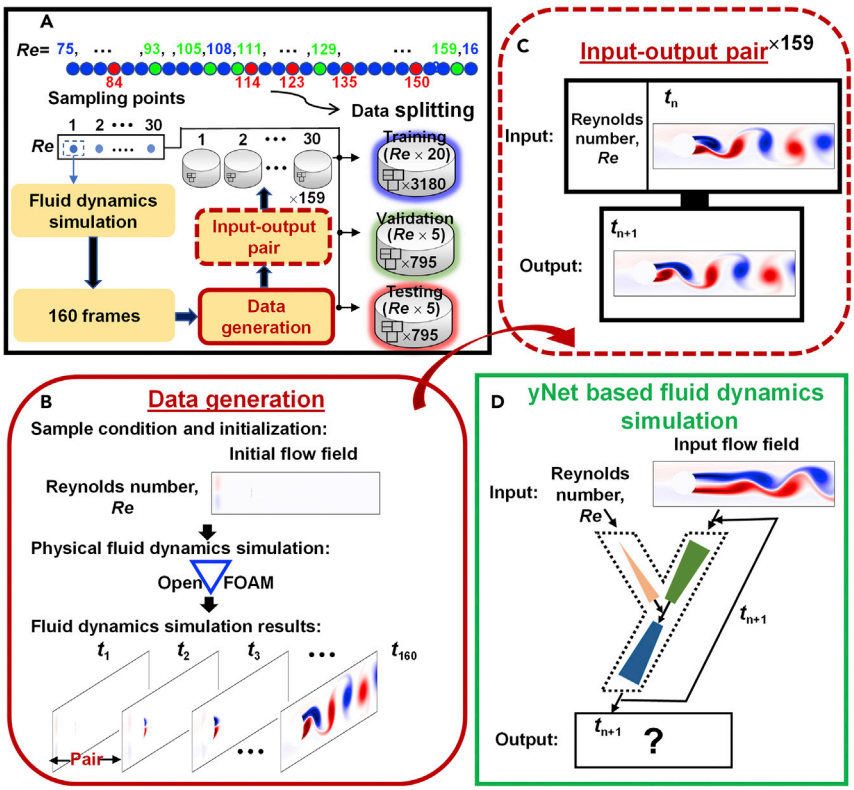


**Figure 2. Illustration of training and testing of yNet for fluid dynamics simulation**

(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 pairs.

(C) Each data point is thus a triplet, consisting of respective Reynolds number and flow fields at $t_n$ and $t_{n+1}$ ($\Delta t$ = 0.1 s), where n = 1, 2, 3, …, 159.

(D) Correspondingly, the trained yNet can be used to predict the flow field for the next time step based on a given flow field and Reynolds number condition. Therefore, the trained yNet can also work in a recursive manner to predict long-term flow field evolution.

commonly used as a benchmark in the data-driven modeling community.[39,40] Specifically, a 2D, incompressible, viscous fluid is studied. In this case, the spatiotemporal evolution of a viscous fluid field is expressed in the non-dimensional form as

$$\frac{\partial \omega'}{\partial t'} = \nabla' \times (\boldsymbol{v}' \times \omega') + \frac{1}{Re}\nabla'^2 \omega', \qquad \text{(Equation 1)}$$

where $\omega'$, $t'$ and $\boldsymbol{v}'$ are vorticity, time, and velocity of fluid in scaled units with respect to unit length $L$ and unit fluid velocity $V$. $Re = \frac{LV}{\nu}$ is the Reynolds number related to kinematic viscosity, unit length, and unit fluid velocity. As implied by Equation 1, the Reynolds number plays a dominant role in determining flow behavior and characterizing the flow pattern.[41] Therefore, 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 characterized by various *Re* unused in the training and validation 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 a conventional multi-input ConvNet.

## Training and testing procedure

Figure 2A illustrates the overall procedure of training an yNet applicable to a range of Reynolds numbers. The transient vorticity field is selected as the flow field of interest and the 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, the yNet is trained to predict a next-step flow field based on the current flow field and *Re* condition. Therefore, a long-term flow field evolution simulation can be performed by yNet through multi-step recursive predictions, as shown in Figure 2D.

More condition parameters can be easily incorporated by simple adjustment of the input layer of MLP, as executed for the next application. To provide a dataset, physical simulations by solving differential equations, including Equation 1, are performed at 30 equally spaced Reynolds numbers (i.e., *Re* = 75, 78, 81, …, 159, 162), where 20 cases would be randomly chosen for training, 5 for validation, and the remaining 5 for testing (see data splitting in Figure 2A). For all physical simulations, extraction of neighboring snapshots is performed to generate one-step-distance data pairs, as illustrated in Figure 2B. A physical simulation will put out a total of 160 frames, indicative of 159 data pairs obtained at each *Re* sampling point or 4,770 for the entire dataset (Figure 2C). They are split into training, validation, and testing datasets based on *Re* as mentioned.

For yNet instantiated in this application, the detailed architecture is illustrated in Figure S1A and explained under Experimental procedures. We train yNet for 100 epochs with a mini-batch size of 2 by using Adam Optimizer with a learning rate of 0.001, $\beta_1$ of 0.9, and $\beta_2$ of 0.999.

## Testing under new *Re* conditions

Figure 3 presents a set of testing results under four different *Re*. As indicated by the predictive error, yNet can accurately predict the next-step flow field in all cases, with a minor difference from the original physics-based simulation. For all 795 testing results, yNet achieves a negligibly small root-mean-square error (RMSE) of 0.426 $s^{-1}$. It represents one of the first cases of data-driven modeling that directly incorporates the Reynolds number. Previous strategies for considering the Reynolds number are usually used to train a neural network that takes as input a series of historical flow fields,[18,42,43] which, as a whole, embed corresponding *Re* information (see LSTM + ConvNet model in Figure S2B). Thus, the neural network can predict the next step based on

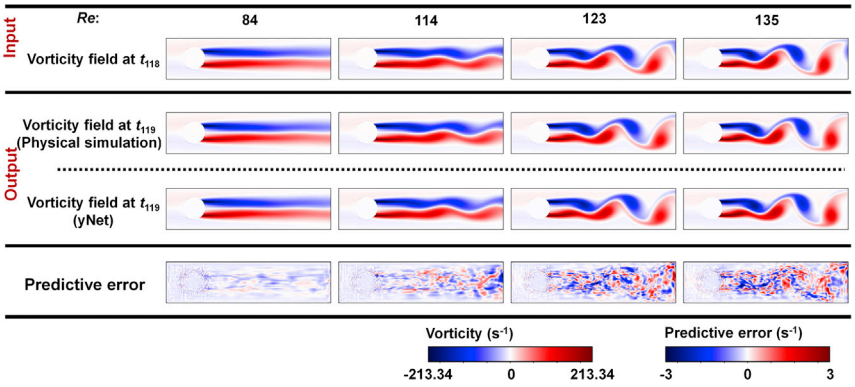


**Figure 3. Testing results of yNet under different *Re* conditions**

We randomly picked a set of testing results that show yNet in predicting next-step flow field development under different *Re* testing conditions.

the evolving trend of consecutive flow fields without being informed by any additional inputs like the Reynolds number. yNet, however, explicitly incorporates the important condition of the Reynolds number and takes as input only one image at the last step, $t_n$. This is to some extent analogous to PDE-based temporal simulation, where the state at $t_{n+1}$ is computed based on that at $t_n$ only instead of a series of historical states. yNet thus possesses two major advantages over previous LSTM + ConvNet strategies.

First, yNet is as conceptually simple as a basic ConvNet without the complexity of integrating a sequence-modeling-purposed LSTM model.[18,42] Second, yNet is widely applicable to general condition-dependent fluid flow simulation problems. One typical example is to predict the development of a flow field for different bluff body shapes.[25,37] This is essentially a different mapping problem between distinct field quantities; i.e., from a structural field (bluff body structure) to a flow field (see comparison in Figure S3). For such regression tasks beyond standard field evolvement, the aforementioned LSTM + ConvNet model apparently will no longer work. We have to explicitly incorporate effects of condition parameters on flow field development by developing a general multi-input field-to-field regression capability.

## Testing on dynamic fluid simulation and temporal extrapolation

Figure 4A shows a representative testing result of predicting multi-step evolution, under the testing condition of *Re* = 84, through recursive inference as depicted in Figure 2D. Noteworthy is that predictive errors can accumulate after many steps of recursive predictions. However, compared with the ground truth of physics-based simulation, yNet well reproduces the long-term dynamic flow field evolution, suggestive of a 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 Figure 4C. The quantitative result reveals that predictive error does accumulate as recursive prediction proceeds but remains at a small level, (i.e., <10.0 $s^{-1}$) for different testing conditions.

Attempts have been made to predict fluid dynamics toward the distant future; i.e., beyond the time domain in training data (see the visualized example for *Re* = 84 in Figure 4B). Correspondingly, for validation purposes, we also extend the physical simulation for another 32 frames. Despite the big challenge associated with temporal extrapolation, predictive error generally continues to display gradual accumulation without a sudden rise, as shown in the right half of Figure 4C. Notably, predictive error can finally reach a maximum RMSE of 19.43 $s^{-1}$ for *Re* = 150. However, such a predictive error is acceptable, still providing good prediction of the flow field, as clearly shown by the inset in Figure 4C. The above results indicate successful learning of the underlying evolution dynamics by yNet, which supports its temporal scalability to reasonably predict flow field evolvement into the distant future.

## yNet versus conventional multi-input ConvNet

Figures 5A and 5B compare the computational accuracy and efficiency, respectively, of yNet and the conventional flattening-based multi-input ConvNet. For fair comparison, both networks are implemented with the exact same encoder and decoder parts (38 layers in total) but using gating- and flattening-based signal-merging strategies respectively. Also, because there are a few variants of the 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 Figure S1. As the validation loss curves in Figure 5A indicate, after reaching convergence of training (>60 epochs), the two models exhibit the same level of validation accuracy. The testing result of the finally trained yNet (RMSE = 0.426 $s^{-1}$) is also comparable with that (RMSE = 0.421 $s^{-1}$) of the conventional multi-input ConvNet. The inset in Figure 5A shows almost the same prediction by the two trained models at a randomly selected testing data point. However, yNet has only one-tenth of the model size of the conventional multi-input ConvNet (yNet, 7,907,685; conventional, 74,900,837) and thus has various computational benefits compared with the conventional ConvNet, such as significantly reduced memory consumption, shorter training time, and faster inference speed (Figure 5B).

These merits of yNet are attributed to the 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 lies in the high expressiveness and meaningfulness of encoded feature maps, which are widely recognized in ConvNet visualization and interpretation studies in the CV community. That is, a feature map would strongly correlate with a certain internal feature/pattern if one projected them back to a pixel-based image.[44,45] Thus, direct manipulation of these feature maps will effectively generate

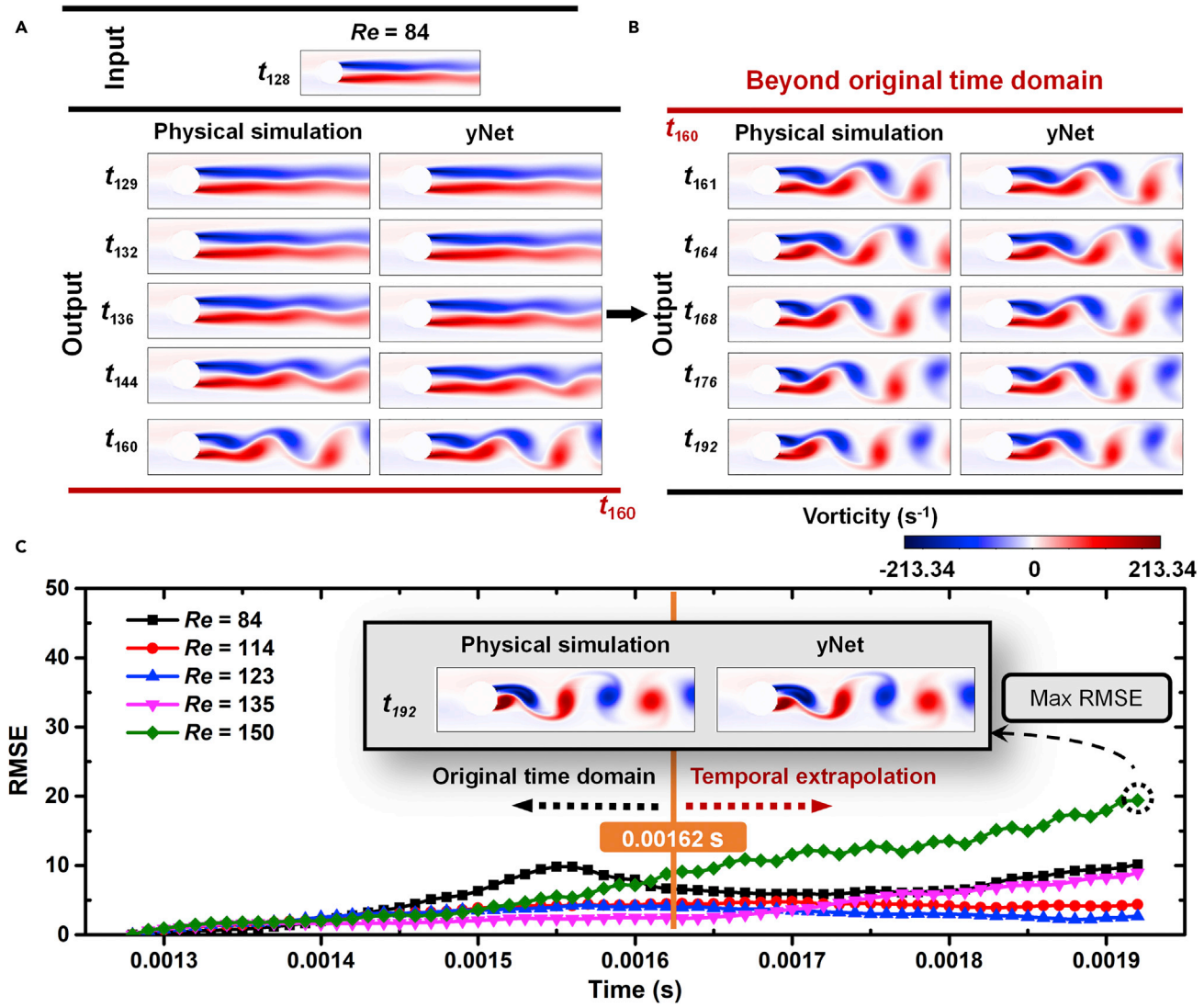


**Figure 4. Testing results of yNet for dynamic fluid simulation**

(A) Testing of yNet in predicting dynamic flow field evolution with time through recursive predictions. In the dynamic simulation, we start from step 128, which is around the wake of unstable flow showing significant evolution. We present snapshots at some time points to 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.

newly weighted combinations of features toward the target image. This is, in spirit, not inconsistent with proper orthogonal decomposition (POD),[46] by which different high-dimensional images can be expressed as different linear combinations of orthogonal modes; i.e., extracted features. However, POD relies on a linear-theory-based feature extraction, which may impair the expressiveness of POD modes.[47,48] In addition, the deep decoder of yNet can contribute to accurate reconstruction of the target or output image through the non-linear and complex deconvolution process.

## Porosity evolution in selective laser sintering

Here we 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 a physics-based model. SLS[49] 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.[50] Fast prediction of porosity formation in a 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 calescence, grain growth by boundary migration through various diffusion mechanisms, and densification (Figure S4).[51] Physics-based SLS simulation is thus usually restricted to a few layers with limited length, far from SLS practice. There is an urgent need for an inexpensive surrogate for the cumbersome physics-based SLS model.

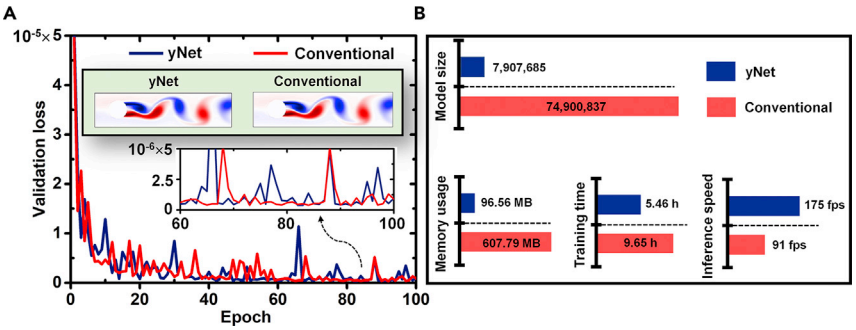


**Figure 5. Comparison of yNet with conventional multi-input ConvNet**

(A and B) Validation loss curves of yNet and conventional multi-input ConvNet (A) and comparison of model size (i.e., total number of parameters) (B) and three other performance metrics between yNet and conventional multi-input ConvNet. The performance is measured purposely based on a laptop (Intel Core i7-7500U CPU, NVIDIA GeForce GTX 950 M GPU, 16 G RAM). Memory use is measured with a single unit input instead of a batch.

## Training and testing procedure

A multi-physical SLS model[52] is used to generate training, validation, and testing datasets. In current SLS modeling, the sintering effects (particle calescence, grain growth, and porosity evolution) are largely decided by the heat-affected zone (HAZ), which depends on the laser condition, and inter-layer interaction, which mainly includes re-sintering of previous layers during scanning of the newly added layer in the layer-by-layer fabrication. It is assumed that the sintering effect (i.e., sintering strength as a main function of laser conditions) 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 this above assumption, we will train yNet based on dataset of small 128 × 128 patches (greater than the maximum HAZ depth of 117 pixels) and then demonstrate its spatial scalability to handling multiple layers of long tracks. Datasets of standard patches are obtained by cropping as-received simulation results every 10 pixels (Figure 6A). The detailed training and testing procedure is illustrated in and Figures S5 and S6 and is described in Note S1, and the detailed yNet architecture implemented in this application is illustrated in Figure S7.

Figure 6A illustrates the data generation of standard patches for training and testing from raw simulation results. The generated dataset is split into training, validation, and testing parts based on laser power and scanning speed sample points; i.e., 75 [*P*, *V*] conditions randomly selected for training, 25 for validation, and the other 25 for testing (see detailed partition of [*P*, *V*] points in Figure S6), and Figure 6B shows 5 testing results on standard patches. yNet predictions closely resemble physics-based simulations, and only a minor morphological error (see last row of Figure 6B) can be observed through pixel-wise comparison of their phase variables. In addition to visual resemblance, we calculate the pixel-wise global accuracy (i.e., the percentage of correct pixels) for such two-phase, high-contrast fields. yNet achieves a similarity as high as 99.53% to ground truth on average for all 43,500 testing results.

## Testing on long tracks and spatial extrapolation

The as-trained yNet is scalable to modeling multiple layers of long tracks. To test step by step, we first examine the spatial extrapolation of the trained yNet to handling a single-layer long track (Figure 7A, and 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 when 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 second step is testing of sintering multi-layer long tracks that involves inter-layer interaction. Figure S8 shows an example of a detailed simulation process for 3 layers. By adopting such a layer-wise simulation, we obtain the multi-layer sintering results by yNet and a physics-based model under different laser conditions (Figure 7C). By increasing laser power, we can see that the results by yNet and the physics-based model both show a transition from the layer-scale unsintered region (Figure 7C, top left inset) to small scattered unsintered areas (Figure 7C, bottom left inset) and, eventually, to a fully sintered state (Figure 7C, bottom right inset). The unsintered regions are located between layers because of insufficient sintering depth under small laser powers. The detailed labeling results of unsintered regions can be also found in Figure S9. The quantitative results of measured unsintered regions (Figure 7C, top right) further prove the good agreement between yNet-based and physical multi-layer simulations.

The tested spatial scalability can help address the computationally daunting task of large-scale sintering simulation, even including a full-component SLS simulation. To demonstrate this, we perform an extreme-scale sintering simulation of a practical 315-layer SLS component with dimensions of 70.8 × 44.1 $mm^2$ (or 3,5416 × 22,050 pixels), as illustrated in Figures S10 and S11 and described in Note S2.

## yNet versus physics-based model

As discussed earlier, physics-based SLS simulation, especially at the component level, is notoriously computationally demanding. It is therefore an ideal case to demonstrate the computational superiority of yNet over a physics-based model. We look into the computation time of yNet and the original physics-based SLS model. Again, yNet and the physics-based model are intentionally tested on a laptop used in the benchmark study in application 1.

Figure 8 shows the computation time as a function of the length of the modeled powder bed. Simulation of small tracks, like the 640-pixel-long tracks in Figure 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 the physics-based model stops at a length of 2,048 pixels because it can be predicted that, by modeling an even longer track, the computation time will increase rapidly to days. For the 8 different lengths tested for both models, yNet is generally 5–6 orders of magnitude faster than the physics-based model; see Table S1

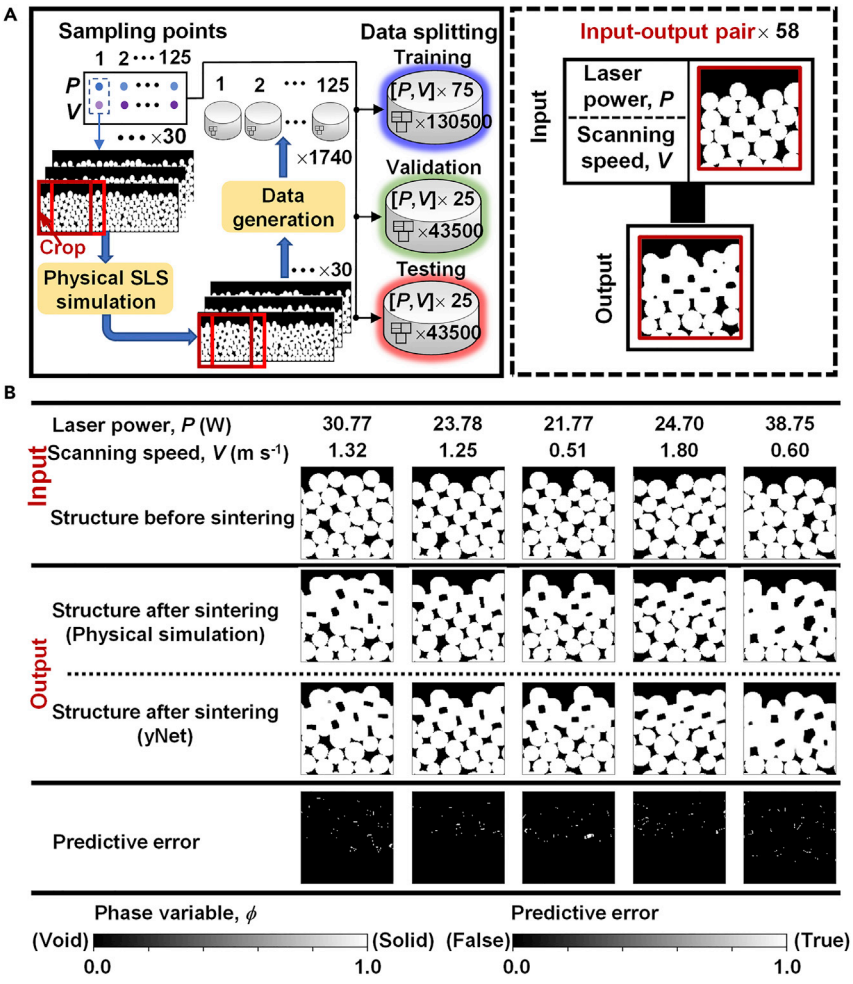


**Figure 6. Testing of yNet on small standard patches**

(A) Pre-processing of raw simulation results for a dataset of standard patches.

(B) Testing results of yNet on standard patches. We randomly selected 5 of 43,500 testing results.

for a more detailed quantitative comparison. yNet can easily handle extremely large layers with a maximum length of ~21,248 pixels with such a limited computational resource. Modeling even this large layer remains nearly instant for yNet. In this regarding, the component-level SLS simulation (i.e., hundreds of large layers), which previously was a formidable task even for high-end computing facilities, can now be accomplished in a few minutes with yNet by simply using a laptop.

## Stress field development

Here we present another application of yNet: predicting stress field development. The main objective of this application is to test the robustness of yNet in (1) handling an input and output field with striking differences 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 with the previous two. The task in this application is to construct mapping between fields of different physical meaning: a structural field and a stress field. Compared with standard field evolvement of a certain type of field, 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 by complex texture and great variability. Therefore, this application may push the performance limit of the proposed yNet in the tough application scenario of strong "field evolvement."

## Training and testing procedure

For illustration purposes, we trained yNet to predict stress field development in a typical perforation structure (i.e., a solid structure with a hole) subjected to compressive loading (Figure S13). Note S3 describes the detailed training and testing procedure. As briefly illustrated in Figure 9A, the basic idea is to train yNet with specific types of perforation structures that feature elliptic holes but with random combinations of hole orientation, size, and aspect ratio. When trained, yNet is expected to predict stress field development for perforation structures with holes of not only elliptic shape but also other geometries.

We first routinely test yNet for perforation structures with elliptical holes unused during training; see Figure S14 for more details. Briefly, the prediction by yNet overall agrees well with the finite element (FE)-based simulation. The RMSE for all 1,000 testing results is as small as 12.70 MPa. Therefore, although the input and output images are totally different and display huge changes, yNet can correctly predict these significant "evolvements."

## Testing on other types of perforation structures and shape extrapolation

Besides routine testing, we tested yNet for perforation structures with other categories of holes, including triangular, rectangular, and polygonal ones, as illustrated in Figure 9A. For the 5 random testing results shown in Figure 9B, the RMSEs are 20.97, 50.43, 49.66, 41.33, and 51.71 MPa, respectively, which are relatively higher than that of testing with elliptic holes. Although the predictive performance shows a slight decrease, from the comparison of detailed stress distribution, yNet still reasonably predicted 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 the fixed boundary at the bottom in the current modeling); the discontinuous geometrical changes transverse to the loading direction, however, do not serve as significant stress risers. These rules are fully contained in the training dataset and universally valid

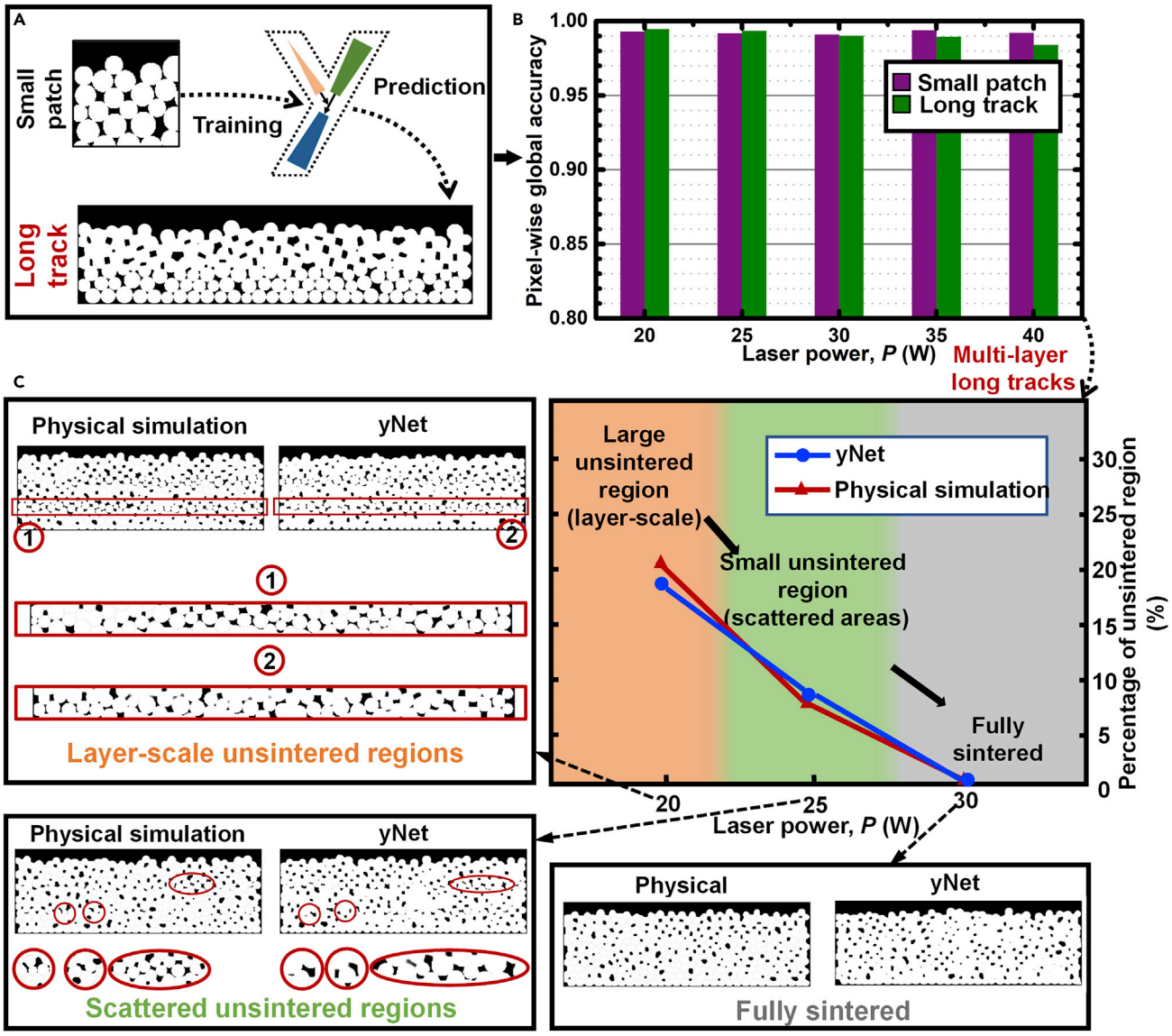


**Figure 7. Testing of yNet on extrapolating prediction for long tracks**

(A) Schematic of the as-trained yNet for predicting sintering of a long track.

(B) Testing results of yNet for predicting sintering of a long track under different laser conditions.

(C) Extension of yNet to predicting sintering of 3-layer long tracks under different laser conditions. By following the workflow in Figure S8, we perform sintering simulation of multiple layers under three laser power conditions (power = 20, 25, and 30 W). Other settings are the same as that used in Figure S8. The overall sintering effect of multiple layers (measured by the percentage of unsintered region) as a function of laser power is then obtained for yNet and the physics-based model, as shown.

for perforation structures of any type. Learning these fundamental rules endows yNet with great extrapolation capability for coping with different perforation structures.

## DISCUSSION

The proposed yNet should be generally applicable to many condition-dependent field evolvement and development problems. Note S4 describes another interesting application of yNet for grain growth simulation; see Figure S15 for the detailed training and testing processes and Figure S16 for some testing results. In addition, we utilize the spatial and temporal scalability of yNet to enable long-term, large-scale grain growth simulation; see Videos S1–S3 for simulations performed at different time resolutions. The four presented applications (3 major and 1 supplemental) have helped us examine the general effectiveness of yNet in handling many distinct field evolvement and development problems. As a general-purpose regression tool, the well-tested yNet should have an enormous potential for future applications. Some promising applications would be ultrafast modeling of structural/morphological evolution in materials science and engineering,[53–55] temperature field development in

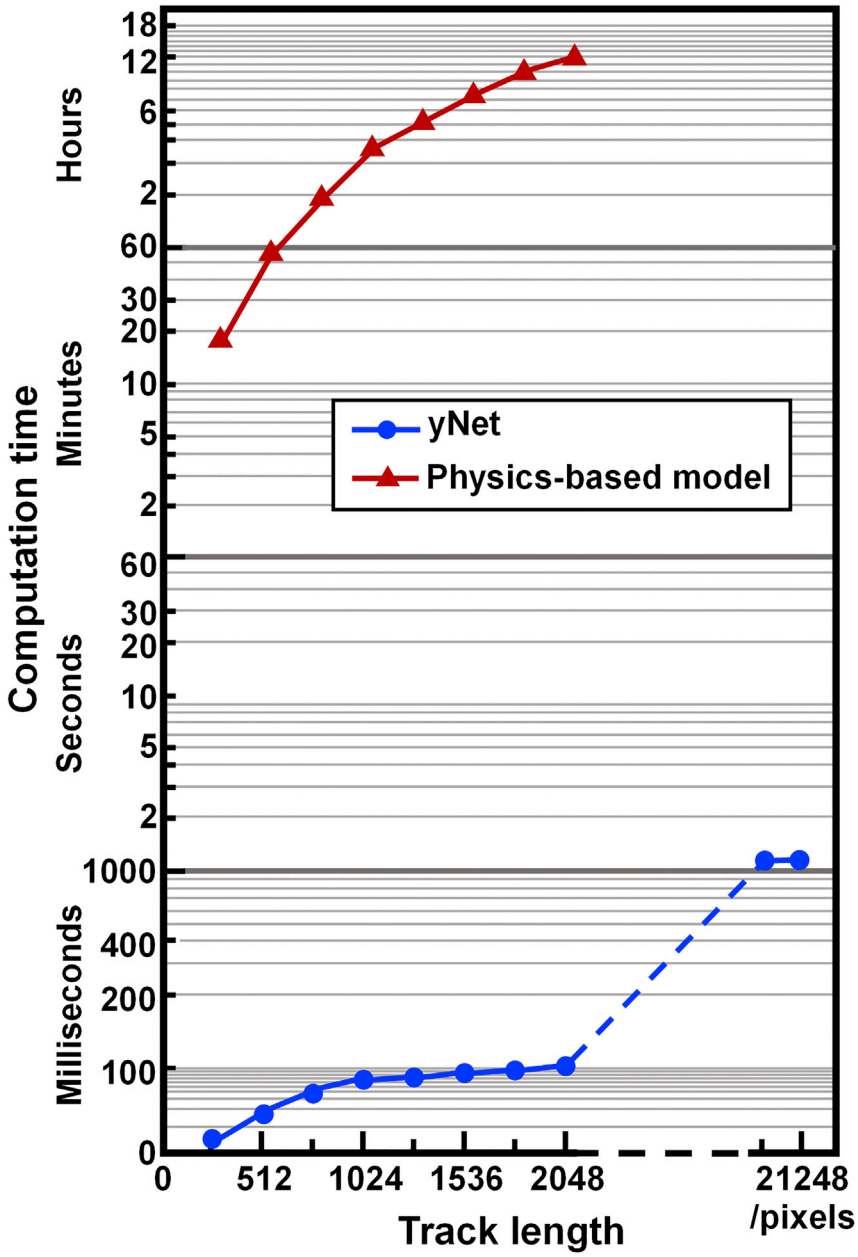


**Figure 8. Comparison of computation time of yNet and a physics-based model**

yNet is not only computationally much faster than the physics-based model but also able to handle extremely long tracks with limited computational resources.

thermal design and control,[56] and geometrical distortion in manufacturing,[57] to name a few.

Physics-informed ML[58] is now emerging as a revolutionary technique by which physics constraints can be imposed to learning processes by directly formulating physical differential equations into loss function[59] or casting them into an energy minimization framework that naturally fits in the loss minimization task of ML training.[60] In doing so, ML would be penalized for its violation of physical laws and forced to learn physical governing equations. Anitescu et al.[61] adopted adaptive collocation to selectively refine training grids at regions with large residuals, which can significantly reduce errors of physics-informed ML. With the rapid advancement of physics-informed ML, the large datasets required for training yNet will be potentially alleviated or even completely eliminated[62] with increased robustness and generalization.

In the future, the extension to real-world data is crucial to fully showcase the modeling capability and realize the practical significance of yNet. Nowadays, data-driven modeling of field evolvement and development still relies principally on simulation-derived datasets because performing highly controlled experiments and measuring 2D or even 3D physical fields with the desired resolution is not only costly but also often technically challenging. However, there have recently been advances in obtaining experimental data at high resolution and even in real time for some physical phenomena. For example, in the fluidics community, besides continued advancement of experimental imaging technologies,[63,64] data postprocessing techniques have been explored to obtain high-resolution flow fields, in time,[65,66] space,[67,68] or both,[69] from as-measured imperfect data. In AM, X-ray computed tomography (XCT) has been increasingly used for characterizing 3D porosity.[70–72] In experimental mechanics, the digital volume correlation (DVC) method permits characterization of 3D displacement, strain, and stress fields with ever-increasing spatiotemporal resolution because of parallel improvement of DVC algorithms and internal imaging instruments.[73–75] It is thus expected that yNet will be readily applied to experimental data in these fields, with fast development of data acquisition techniques and upon availability of (sufficient) high-quality experimental data in the near future.

The main limitation of yNet is that the trained yNet is sometimes limited in scaling to fields 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 the applied laser condition and is consistent along the length of the sintered track for a given laser condition (see the illustration in Figure S12). The large-scale simulation of natural grain growth is also based on its completely uniform evolving kinetics over the modeling

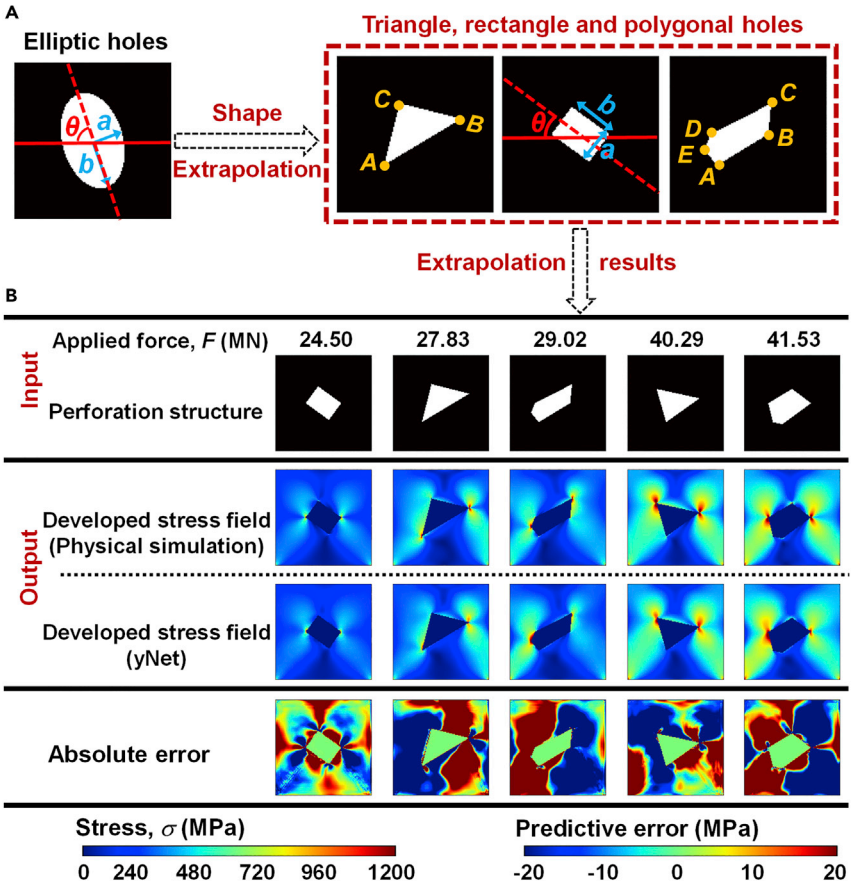


**Figure 9. Testing of yNet on predicting stress development for various types of perforation structures**

(A) yNet is trained based on a simple dataset containing elliptical holes only. The as-trained yNet can predict stress fields for other types of holes. All holes used in training, validation, and testing are randomly generated based on the descriptive methods as shown.

(B) Testing of yNet on different types of perforation structures. The perforation structures have completely different types of holes than that used in the training dataset. Five randomly generated rectangular, triangular, and polygonal holes are tested.

domain. This is, however, not always the case; e.g., for fluid dynamics simulation and stress field prediction where such simple scalability may not hold. Other solutions are needed 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 for various physical and engineering problems. For example, the applied loading in stress field modeling in Figure 9 can be non-uniform and, thus, not describable by a single scalar parameter. Also, besides natural grain growth (Note S4), grain growth in practice can be influenced by the temperature of the field because the preferred growth direction is along the local thermal gradient.[76] These complex conditions, no matter how many dimensions they have, can be eventually converted to an embedding vector of the desired length. It would then allow adoption of gating-based signal merging for efficient incorporation of their effects on field evolvement.

We have proposed a multi-input ConvNet, yNet, with stunning simplicity and usefulness, which are concretely demonstrated through comprehensive applications and testing. It thus may solidly push our fundamental regression capability to multi-input field-to-field/image-to-image regression, which is broadly in demand in data-driven modeling across many physical and engineering domains. yNet merges input signals without flattening the field input into a large vector, instead directly manipulating high-level-encoded feature maps. It possesses a significant computational advantage over its ConvNet counterpart and extreme computational efficiency compared with the physics-based model. Based on its computational superiority, large-scale and/or long-term dynamic simulation of field evolvements can be easily performed with even limited computational resources. As indicated by the unprecedented wide and deep applications presented in this research, the proposed yNet may have enormous potential and a profound effect as a conceptually simple, lightweight, but powerful multi-input field-to-field regression tool.

## EXPERIMENTAL PROCEDURES

### Resource availability

#### *Lead contact*

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, L.C. (leichn@umich.edu).

#### *Materials availability*

This study did not generate new unique materials or reagents.

#### *Data and code availability*

All datasets reported in this study have been deposited at Mendeley Data: http://doi.org/10.17632/gg8f2hwkxd.1, as Four Datasets for Multi-input ConvNet. All datasets reported in this paper will also be readily shared by the lead contact upon request. All original code has been deposited at Zenodo (http://doi.org/10.5281/zenodo.6370878) and in GitHub (https://github.com/zwang586/MICNN). Any additional information required to reanalyze the data reported in this paper is available from the lead contact upon request.

### Fluid dynamics model

A two-dimensional, incompressible, viscous model is used to study flow past a cylinder using the numerical method in OpenFOAM. The full governing equations in describing the flow are

$$\nabla \cdot \boldsymbol{v} = 0 \qquad \text{(Equation 2)}$$

$$\frac{\partial \boldsymbol{v}}{\partial t} + \boldsymbol{v} \cdot \nabla \boldsymbol{v} = -\frac{1}{\rho}\nabla p + \nu \nabla^2 \boldsymbol{v}, \qquad \text{(Equation 3)}$$

where $\boldsymbol{v}$ is the velocity field, $p$ is the pressure field, and $\nu$ and $\rho$ are the kinematic viscosity and density of the fluid, respectively.

The discretized continuity and Navier-Stokes Equations 2 and 3 are solved using the pressure implicit with splitting of operator (PISO) algorithm on a mesh consisting of 85,720 cells. Specifically, the computational fluid dynamics (CFD) simulation is realized using icoFoam. The velocity field is computed into the vorticity field for flow visualization. It is important to transfer the Navier-Stokes equation into the non-dimensional form so that a dimensionless number is used to characterize the system. Hereby, the Reynolds number is introduced using dimensional analysis[41] that represents the ratio of inertial forces to the viscous forces in the flow field, as shown in Equation 1. In this case, variables in Equations 2 and 3 can be rewritten in scaled units with respect to typical length $L$ (diameter of the cylinder) and typical fluid velocity $V$ (incoming flow velocity); namely, $x = x'L$, $\boldsymbol{v} = \boldsymbol{v}'V$, $t = t'\frac{L}{V}$, $\omega = \omega'\frac{V}{L}$. We focus on the moderate Reynolds number range where a wake is developed passing a cylinder and oscillating with a certain frequency. The flow behavior is identified as a von Karman vortex and is ideal to characterize the flow pattern with different Reynold numbers.

## SLS model

In this study, we basically customize a phase-field-based sintering model[51] for applicability in SLS, with incorporation of a heat transfer model and a Gaussian heat source model describing the effective heat input from a moving laser beam.[77] The sintering model has properly taken into account multiple physical processes by first reformulating the effective diffusion coefficient as

$$D^{eff} = D_{vol}\varphi(\rho) + D_{vap}[1 - \varphi(\rho)] + D_{sf}\rho(1 - \rho) + D_{gb}\sum_{\alpha}\sum_{\alpha' \neq \alpha}(\eta_\alpha \eta_{\alpha'}), \quad \text{(Equation 4)}$$

where, $D_{vol}$, $D_{vap}$, $D_{sf}$, and $D_{gb}$ are temperature-dependent diffusion coefficients in solid volume, vapor, along the surface, and the grain boundary, respectively, which are thus dependent on fully coupled thermal modeling. $\eta_\alpha$ is phase variable that describes the $\alpha$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, $\boldsymbol{v}_{rbm} = \sum_{\alpha}[\boldsymbol{v}_t(\alpha) + \boldsymbol{v}_r(\alpha)]$. The velocity fields of rigid-body translation, $\boldsymbol{v}_t$ and rotation, $\boldsymbol{v}_r$, of the $\alpha$th particle are calculated by

$$\boldsymbol{v}_t(\boldsymbol{r}, \alpha) = \frac{m_t}{V(\alpha)}\boldsymbol{F}(\alpha)\eta_\alpha(\boldsymbol{r}) \quad \text{(Equation 5)}$$

$$\boldsymbol{v}_r(\boldsymbol{r}, \alpha) = \frac{m_r}{V(\alpha)}\boldsymbol{T}(\alpha)[\boldsymbol{r} - \boldsymbol{r}_c(\alpha)]\eta_\alpha(\boldsymbol{r}), \quad \text{(Equation 6)}$$

where $m_t = 500$ and $m_r = 1$. The force, $\boldsymbol{F}$, and torque, $\boldsymbol{T}$, acting on the powder particles are associated with the their real-time structural environment and materials density. Full mathematical details of our physical SLS model are reported in a separate paper on multi-physical SLS simulation.[52]

## Powder generation model

In addition to the sintering model for simulating porosity evolution, generation of a powder bed (i.e., the initial porous structure) is simulated using a "rain" model.[78] The basic mechanism of a 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 previous 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 radius of deposited powders are 25 and 2.5 μm, respectively. It should be pointed out that yNet in this study will be trained to simulate porosity evolution and thus replace a physics-based sintering model only. The powder bed generation model is a separate model that provides an initial structure for physical sintering simulation and yNet-based sintering simulation.

## Solid mechanics model

An FE-based solid mechanics model is used to generate stress field datasets. The modeled object is a perforation structure with dimensions of 0.1 × 0.1 $m^2$. To generate random perforation structures, the elliptical hole at the center is defined by the length of a-semiaxis and b-semiaxis as well as rotation angle. The rectangular hole is also located at the center and defined by width, height, and rotation angle. The triangular hole and polygonal hole are specified by the coordinates of their respective vertices. Fixed constraint is applied at the bottom of the perforation structure and uniform compressive force at the top. This model is implemented in COMSOL Multiphysics 5.5.

## Grain growth model

A phase-field grain growth model is used to perform natural grain growth simulation and generate grain structure datasets. In the phase-field model, each grain is described by a phase variable, $\varphi_q$. The grain growth is then described by the temporal and spatial evolution of the phase variable, which can be simulated by solving the time-dependent Ginzburg-Landau equation

$$\frac{\partial\varphi_q(\boldsymbol{r}, t)}{\partial t} = -L\frac{\delta F(t)}{\delta\varphi_q(\boldsymbol{r}, t)}, \quad \text{(Equation 7)}$$

where $L$ is the kinetic rate coefficient related to the grain boundary mobility and $F$ is the total free energy of a polycrystalline system that drives the microstructure evolution.[79] All phase-field grain growth simulations start with a random seeding of 150 grains of size $r = 5$. A periodic boundary condition is applied for the top (left) and bottom (right) boundary.

## yNet

We use the yNet instantiated in fluid dynamics simulation as an illustrative example. The detailed architecture is shown in Figure 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 has 512 neurons. For the encoder, the input is a vorticity flow field at the current step in a 256 × 64 image. Each green block represents a combination of Conv + Relu operations, and a light green block means max-pooling. In this manner, the encoder finally yields 512 8 × 2 feature maps, which are passed to a dropout layer with a rate of 0.5 before merging with the MLP signal. The decoder just has a somewhat mirrored topology of the encoder, with each blue block representing a combination of Conv + Relu operations and dark blue block up-sampling. At the end of the MLP and encoder, conventional approaches adopted a flattening-based strategy to merge multi-input signals. As illustrated in Figure 1E, the signal merging through a sequence of flattening, concatenating, compressing, and reshaping operations can be expressed as

$$\boldsymbol{U}_{en} \rightarrow \boldsymbol{s}_{en},\ \boldsymbol{U}_{en} \in \mathbb{R}^{n_1 \times n_2 \times N_2}, \boldsymbol{s}_{en} \in \mathbb{R}^{N_3} \quad \text{(Equation 8)}$$

$$\boldsymbol{s}_{concatenate} = \boldsymbol{s}_{MLP} \odot \boldsymbol{s}_{en},\ \boldsymbol{s}_{concatenate} \in \mathbb{R}^{N_1 + N_3}, \boldsymbol{s}_{MLP} \in \mathbb{R}^{N_1} \quad \text{(Equation 9)}$$

$$\boldsymbol{s}_{concatenate} \rightarrow \boldsymbol{s}_{compress},\ \boldsymbol{s}_{compress} \in \mathbb{R}^{N_3} \quad \text{(Equation 10)}$$

$$\boldsymbol{s}_{compress} \rightarrow \boldsymbol{U}_{de},\ \boldsymbol{U}_{de} \in \mathbb{R}^{n_1 \times n_2 \times N_2}, \quad \text{(Equation 11)}$$

where $n_1 \times n_2$ and $N_2$ are, respectively, the shape and number of encoded feature maps, $\boldsymbol{U}_{en}$ at the end of encoder, $N_3 = n_1 \times n_2 \times N_2$ is the dimension of the vector, $\boldsymbol{s}_{en}$, by flattening feature maps, $\odot$ denotes concatenate operation, $\boldsymbol{s}_{concatenate}$ is the concatenation result, $N_1$ is the dimension of the output vector of MLP, $\boldsymbol{s}_{MLP}$, $\boldsymbol{s}_{compress}$ represent the resulting vector by compressing $\boldsymbol{s}_{concatenate}$, and $\boldsymbol{U}_{de}$ is finally obtained feature maps through reshaping $\boldsymbol{s}_{compress}$ for decoding process.

For yNet, we propose to simply merge the signal using the gating-based strategy as depicted in Figure 1G,

$$\boldsymbol{U}_{de} = \boldsymbol{s}_{MLP} \circ \boldsymbol{U}_{en}, \quad \text{(Equation 12)}$$

where $\circ$ denotes channel-wise multiplication, and $\boldsymbol{U}_{de}$ is the multiplication results (gated feature maps) for the decoding process. To successfully perform channel-wise multiplication, the number of neurons in the last layer of MLP is always set equal to the number of feature maps at the end of encoder; i.e., $N_1 = N_2$ in Figure 1G.

For the other three applications, we basically use the yNet described above but with the potentially adjusted depth of encoder-decoder and width of MLP depending on the complexity of the modeling problem. Specifically, in data-driven modeling of stress field development, we use the same yNet as implemented in modeling fluid dynamics (Figure S1A). In data-driven modeling of porosity development in SLS and grain growth, we implement a slightly shallower yNet (Figure S7), considering the relative textural simplicity of the images to be processed. Correspondingly, MLP is properly adjusted to guarantee channel-wise multiplication at the end of MLP and ConvNet encoder. In all four applications, min-max normalization of datasets is performed for ease of training.

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

$$Loss = \frac{1}{N}\sum_{i=1}^{N}\left|f\left(\boldsymbol{a}^i, X^i\right) - \overline{Y}^i\right|^2, \quad \text{(Equation 13)}$$

where $N$ is the number of samples processed, $\boldsymbol{a}$ is the condition-parameter input, $X$ is the field input, $f(\boldsymbol{a}, X)$ is the evolved field by prediction, and $\overline{Y}$ is the evolved field of ground truth. For an 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:

$$Loss = -\frac{1}{N}\sum_{i=1}^{N}\left|f\left(\boldsymbol{a}^i, X^i\right)\cdot\log\overline{Y}^i + \left(1 - f\left(\boldsymbol{a}^i, X^i\right)\right)\cdot\left(1 - \log\overline{Y}^i\right)\right|^2. \quad \text{(Equation 14)}$$

In all four applications, yNet is trained end to end by using the training hyperparameters listed in Table S2. Learning curves are summarized in Figure S17.

## SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.patter.2022.100494.

## ACKNOWLEDGMENTS

The research is financially supported by National Science Foundation (NSF) grant CMMI-2020527 and startup funds from the Department of Mechanical Engineering at University of Michigan-Dearborn. The extensive physics simulations were carried out through computational resources and services provided by Advanced Research Computation (ARC) at University of Michigan and High Performance Computing Collaboratory (HPC$^2$) at Mississippi State University.

## AUTHOR CONTRIBUTIONS

Z.W. and W.Y. prepared figures, performed results analysis, wrote the paper, and prepared supplemental information. L.X. and O.Z. conducted extensive fluid flow simulations. W.Y. and Y.L. carried out extensive physics-based SLS simulations. X.W. helped with developing the multi-physical porosity simulation model. Y.Z. and P.L. performed extensive grain growth simulations. Z.W. performed massive stress field development simulations. Y.X. contributed to powder bed generation in full-component porosity simulation. M.B. contributed by consulting and designing the SLS part. L.C. conceived the application of convolutional network to simulating field evolvements and supervised this research. All authors discussed the results and commented on the manuscript.

## DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: December 8, 2021
Revised: February 18, 2022
Accepted: March 24, 2022
Published: April 21, 2022

## REFERENCES

1. LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. Nature *521*, 436–444.
2. LeCun, Y., Boser, B.E., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W.E., and Jackel, L.D. (1990). Handwritten digit recognition with a back-propagation network. In Advances in Neural Information Processing Systems, pp. 396–404.
3. Rawat, W., and Wang, Z. (2017). Deep convolutional neural networks for image classification: a comprehensive review. Neural Comput. *29*, 2352–2449.
4. Du, G., Wang, K., and Lian, S. (2019). Vision-based robotic grasping from object localization, pose estimation, grasp detection to motion planning: a review. Preprint at arXiv. https://doi.org/10.48550/arXiv.1905.06658.
5. Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., and Garcia-Rodriguez, J. (2017). A review on deep learning techniques applied to semantic segmentation. Preprint at arXiv. https://doi.org/10.48550/arXiv.1704.06857.
6. Guo, Y., Liu, Y., Georgiou, T., and Lew, M.S. (2018). A review of semantic segmentation using deep neural networks. Int. J. Multimed. Inf. Retr. *7*, 87–93.
7. Ling, J., Hutchinson, M., Antono, E., DeCost, B., Holm, E.A., and Meredig, B. (2017). Building data-driven models with microstructural images: generalization and interpretability. Mater. Discov. *10*, 19–28.
8. DeCost, B.L., and Holm, E.A. (2015). A computer vision approach for automated analysis and classification of microstructural image data. Comput. Mater. Sci. *110*, 126–133.
9. Li, X., Zhang, Y., Zhao, H., Burkhart, C., Brinson, L.C., and Chen, W. (2018). A transfer learning approach for microstructure reconstruction and structure-property predictions. Sci. Rep. *8*, 1–13.
10. Cang, R., Xu, Y., Chen, S., Liu, Y., Jiao, Y., and Ren, M.Y. (2017). Microstructure representation and reconstruction of heterogeneous materials via deep belief network for computational material design. J. Mech. Des. *139*, 071404.
11. Yang, W., Wang, Z., Yang, T., He, L., Song, X., Liu, Y., and Chen, L. (2021). Exploration of the underlying space in microscopic images via deep learning for additively manufactured piezoceramics. ACS Appl. Mater. Inter. *13*, 53439–53453.
12. Imani, F., Chen, R., Diewald, E., Reutzel, E., and Yang, H. (2019). Deep learning of variant geometry in layerwise imaging profiles for additive manufacturing quality control. J. Manuf. Sci. Eng. *141*, 081004.
13. Tian, Q., Guo, S., Melder, E., Bian, L., and Guo, W. (2021). Deep learning-based data fusion method for in situ porosity detection in laser-based additive manufacturing. J. Manuf. Sci. Eng. *143*, 041011.
14. Herriott, C., and Spear, A.D. (2020). Predicting microstructure-dependent mechanical properties in additively manufactured metals with machine- and deep-learning methods. Comput. Mater. Sci. *175*, 109599.
15. Yang, Z., Yabansu, Y.C., Al-Bahrani, R., Liao, W.-k., Choudhary, A.N., Kalidindi, S.R., and Agrawal, A. (2018). Deep learning approaches for mining structure-property linkages in high contrast composites from simulation datasets. Comput. Mater. Sci. *151*, 278–287.
16. Le, T., Epa, V.C., Burden, F.R., and Winkler, D.A. (2012). Quantitative structure–property relationship modeling of diverse materials properties. Chem. Rev. *112*, 2889–2919.
17. Yan, W., Lin, S., Kafka, O.L., Yu, C., Liu, Z., Lian, Y., Wolff, S., Cao, J., Wagner, G.J., and Liu, W.K. (2018). Modeling process-structure-property relationships for additive manufacturing. Front. Mech. Eng. *13*, 482–492.
18. Hasegawa, K., Fukami, K., Murata, T., and Fukagata, K. (2020). CNN-LSTM based reduced order modeling of two-dimensional unsteady flows around a circular cylinder at different Reynolds numbers. Fluid Dyn. Res. *52*, 065501.
19. Yang, K., Cao, Y., Zhang, Y., Fan, S., Tang, M., Aberg, D., Sadigh, B., and Zhou, F. (2021). Self-supervised learning and prediction of microstructure

evolution with convolutional recurrent neural networks. Patterns *2*, 100243.

20. Frankel, A., Tachida, K., and Jones, R. (2020). Prediction of the evolution of the stress field of polycrystals undergoing elastic-plastic deformation with a hybrid neural network model. Mach. Learn. Sci. Technol. *1*, 035005.
21. Raj, M., Thakre, S., Annabattula, R.K., and Kanjarla, A.K. (2021). Estimation of local strain fields in two-phase elastic composite materials using UNet-based deep learning. Integr. Mater. Manuf. Innov. *10*, 444–460.
22. Zhu, Y., and Zabaras, N. (2018). Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. J. Comput. Phys. *366*, 415–447.
23. He, Y., Fei, F., Wang, W., Song, X., Sun, Z., and Baek, S. (2018). 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, V01BT02A033–V001BT002A033.
24. Shen, Z., Shang, X., Zhao, M., Dong, X., Xiong, G., and Wang, F.-Y. (2019). A learning-based framework for error compensation in 3D printing. IEEE Trans. Cybern. *49*, 4042–4050.
25. Guo, X., Li, W., and Iorio, F. (2016). Convolutional neural networks for steady flow approximation. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp. 481–490.
26. Mianroodi, J.R., Siboni, N.H., and Raabe, D. (2021). Teaching solid mechanics to artificial intelligence—a fast solver for heterogeneous materials. Npj Comput. Mater. *7*, 1–10.
27. Croom, B.P., Berkson, M., Mueller, R.K., Presley, M., and Storck, S. (2021). Deep learning prediction of stress fields in additively manufactured metals with intricate defect networks. Preprint at arXiv. https://doi.org/10.48550/arXiv.2105.10564.
28. Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention, pp. 234–241.
29. Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440.
30. Yuan, Z., Jiang, Y., Li, J., and Huang, H. (2020). Hybrid-DNNs: hybrid deep neural networks for mixed inputs. Preprint at arXiv. https://doi.org/10.48550/arXiv.2005.08419. https://arxiv.org/abs/2005.08419.
31. Law, S., Paige, B., and Russell, C. (2019). Take a look around: using street view and satellite images to estimate house prices. ACM Trans. Intell. Syst. Technol. *10*, 1–19.
32. Allaire, F., Mallet, V., and Filippi, J.-B. (2021). Emulation of wildland fire spread simulation using deep learning. Neural Netw. *141*, 184–198.
33. Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). Segnet: a deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. *39*, 2481–2495.
34. Noh, H., Hong, S., and Han, B. (2015). Learning deconvolution network for semantic segmentation. In Proceedings of the IEEE international conference on computer vision, pp. 1520–1528.
35. Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A.L. (2014). Semantic image segmentation with deep convolutional nets and fully connected crfs. Preprint at arXiv. https://doi.org/10.48550/arXiv.1412.7062.
36. Nie, Z., Jiang, H., and Kara, L.B. (2019). Stress field prediction in cantilevered structures using convolutional neural networks. J. Comput. Inf. Sci. Eng. *20*, 011002.
37. Bhatnagar, S., Afshar, Y., Pan, S., Duraisamy, K., and Kaushik, S. (2019). Prediction of aerodynamic flow fields using convolutional neural networks. Comput. Mech. *64*, 525–545.
38. Hu, J., Shen, L., and Sun, G. (2018). Squeeze-and-excitation networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7132–7141.
39. Rudy, S.H., Brunton, S.L., Proctor, J.L., and Kutz, J.N. (2017). Data-driven discovery of partial differential equations. Sci. Adv. *3*, e1602614.
40. Kutz, J.N., Brunton, S.L., Brunton, B.W., and Proctor, J.L. (2016). Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems (SIAM).
41. Kundu, P.K., Cohen, I.M., and Dowling, D.R. (2015). Fluid Mechanics (Academic press).
42. Hasegawa, K., Fukami, K., Murata, T., and Fukagata, K. (2020). Machine-learning-based reduced-order modeling for unsteady flows around bluff bodies of various shapes. Theor. Comput. Fluid Dyn. *34*, 367–383.
43. Lee, S., and You, D. (2017). Prediction of laminar vortex shedding over a cylinder using deep learning. Preprint at arXiv. https://doi.org/10.48550/arXiv.1712.07854.
44. Zeiler, M.D., and Fergus, R. (2014). Visualizing and understanding convolutional networks. In European conference on computer vision, pp. 818–833.
45. Qin, Z., Yu, F., Liu, C., and Chen, X. (2018). How convolutional neural network see the world-A survey of convolutional neural network visualization methods. Preprint at arXiv. https://doi.org/10.48550/arXiv.1804.11191.
46. Chatterjee, A. (2000). An introduction to the proper orthogonal decomposition. Curr. Sci. *78*, 808–817.
47. Baldi, P., and Hornik, K. (1989). Neural networks and principal component analysis: learning from examples without local minima. Neural Netw. *2*, 53–58.
48. Murata, T., Fukami, K., and Fukagata, K. (2020). Nonlinear mode decomposition with convolutional neural networks for fluid dynamics. J. Fluid Mech. *882*, A13.
49. Kumar, S. (2003). Selective laser sintering: a qualitative and objective approach. JOM *55*, 43–47.
50. Dürr, H., Pilz, R., and Eleser, N.S. (1999). Rapid tooling of EDM electrodes by means of selective laser sintering. Comput. Ind. *39*, 35–45.
51. Wang, Y.U. (2006). Computer modeling and simulation of solid-state sintering: a phase field approach. Acta Mater. *54*, 953–961.
52. Wang, X., Liu, Y., Li, L., Yenusah, C.O., Xiao, Y., and Chen, L. (2021). Multiscale phase-field modeling of layer-by-layer powder compact densification during solid-state direct metal laser sintering. Mater. Des. *203*, 109615.
53. Chen, L., Zhang, H.W., Liang, L.Y., Liu, Z., Qi, Y., Lu, P., Chen, J., and Chen, L.-Q. (2015). Modulation of dendritic patterns during electrodeposition: a nonlinear phase-field model. J. Power Sourc. *300*, 376–385.
54. Wang, X., Liu, P., Ji, Y., Liu, Y., Horstemeyer, M., and Chen, L. (2019). Investigation on microsegregation of IN718 alloy during additive manufacturing via integrated phase-field and finite-element modeling. J. Mater. Eng. Perform. *28*, 657–665.
55. Chen, L.-Q. (2002). Phase-field models for microstructure evolution. Annu. Rev. Mater. Res. *32*, 113–140.
56. Wang, Z., Li, Q., Trinh, W., Lu, Q., Cho, H., Wang, Q., and Chen, L. (2017). Optimal design of high temperature metalized thin-film polymer capacitors: a combined numerical and experimental method. J. Power Sourc. *357*, 149–157.
57. Paul, R., Anand, S., and Gerner, F. (2014). Effect of thermal deformation on part errors in metal powder based additive manufacturing processes. J. Manuf. Sci. Eng. *136*, 031009.
58. Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. (2021). Physics-informed machine learning. Nat. Rev. Phys. *3*, 422–440.
59. Raissi, M., Perdikaris, P., and Karniadakis, G.E. (2019). Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. *378*, 686–707.

60. Samaniego, E., Anitescu, C., Goswami, S., Nguyen-Thanh, V.M., Guo, H., Hamdia, K., Zhuang, X., and Rabczuk, T. (2020). An energy approach to the solution of partial differential equations in computational mechanics via machine learning: concepts, implementation and applications. Comput. Methods Appl. Mech. Eng. *362*, 112790.

61. Anitescu, C., Atroshchenko, E., Alajlan, N., and Rabczuk, T. (2019). Artificial neural network methods for the solution of second order boundary value problems. Comput. Mater. Contin. *59*, 345–359.

62. Zhu, Y., Zabaras, N., Koutsourelakis, P.-S., and Perdikaris, P. (2019). Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data. J. Comput. Phys. *394*, 56–81.

63. Raffel, M. (2015). Background-oriented schlieren (BOS) techniques. Exp. Fluids *56*, 1–17.

64. Stankovic, Z., Allen, B.D., Garcia, J., Jarvis, K.B., and Markl, M. (2014). 4D flow imaging with MRI. Cardiovasc. Diagn. Ther. *4*, 173–192.

65. Cai, S., Wang, Z., Fuest, F., Jeon, Y.J., Gray, C., and Karniadakis, G.E. (2021). Flow over an espresso cup: inferring 3-D velocity and pressure fields from tomographic background oriented Schlieren via physics-informed neural networks. J. Fluid Mech. *915*, A102.

66. Krishna, C.V., Wang, M., Hemati, M.S., and Luhar, M. (2020). Reconstructing the time evolution of wall-bounded turbulent flows from non-time-resolved PIV measurements. Phys. Rev. Fluids *5*, 054604.

67. Ferdian, E., Suinesiaputra, A., Dubowitz, D.J., Zhao, D., Wang, A., Cowan, B., and Young, A.A. (2020). 4DFlowNet: super-resolution 4D flow MRI using deep learning and computational fluid dynamics. Front. Phys. *8*, 138.

68. Gao, H., Sun, L., and Wang, J.-X. (2021). Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels. Phys. Fluids *33*, 073603.

69. Jin, X., Laima, S., Chen, W.-L., and Li, H. (2020). Time-resolved reconstruction of flow field around a circular cylinder by recurrent neural networks based on non-time-resolved particle image velocimetry measurements. Exp. Fluids *61*, 1–23.

70. Cunningham, R., Narra, S.P., Montgomery, C., Beuth, J., and Rollett, A. (2017). Synchrotron-based X-ray microtomography characterization of the effect of processing variables on porosity formation in laser power-bed additive manufacturing of Ti-6Al-4V. JOM *69*, 479–484.

71. Shrestha, S., Starr, T., and Chou, K. (2019). A study of keyhole porosity in selective laser melting: single-track scanning with micro-CT analysis. J. Manuf. Sci. Eng. *141*, 071004.

72. King, W.E., Barth, H.D., Castillo, V.M., Gallegos, G.F., Gibbs, J.W., Hahn, D.E., Kamath, C., and Rubenchik, A.M. (2014). Observation of keyhole-mode laser melting in laser powder-bed fusion additive manufacturing. J. Mater. Process. Technol. *214*, 2915–2925.

73. Xu, F. (2018). Quantitative characterization of deformation and damage process by digital volume correlation: a review. Theor. Appl. Mech. Lett. *8*, 83–96.

74. Pan, B. (2018). Digital image correlation for surface deformation measurement: historical developments, recent advances and future goals. Meas. Sci. Technol. *29*, 082001.

75. Buljac, A., Jailin, C., Mendoza, A., Neggers, J., Taillandier-Thomas, T., Bouterf, A., Smaniotto, B., Hild, F., and Roux, S. (2018). Digital volume correlation: review of progress and challenges. Exp. Mech. *58*, 661–708.

76. Liu, P., Ji, Y., Wang, Z., Qiu, C., Antonysamy, A., Chen, L.-Q., Cui, X., and Chen, L. (2018). Investigation on evolution mechanisms of site-specific grain structures during metal additive manufacturing. J. Mater. Process. Technol. *257*, 191–202.

77. Yang, Y., Kühn, P., Yi, M., Egger, H., and Xu, B.-X. (2020). Non-isothermal phase-field modeling of heat–melt–microstructure-coupled processes during powder bed fusion. JOM *72*, 1719–1733.

78. Lu, L.-X., Sridhar, N., and Zhang, Y.-W. (2018). Phase field simulation of powder bed-based additive manufacturing. Acta Mater. *144*, 801–809.

79. Krill, C., Iii, and Chen, L.-Q. (2002). Computer simulation of 3-D grain growth using a phase-field model. Acta Mater. *50*, 3059–3075.