Computational Materials Science 213 (2022) 111599

Contents lists available at ScienceDirect

Computational Materials Science

journal homepage: www.elsevier.com/locate/commatsci

Full length article

Multi-fidelity machine learning models for structure–property mapping of organic electronics

Chih-Hsuan Yang [a], Balaji Sesha Sarath Pokuri [a], Xian Yeow Lee [a], Sangeeth Balakrishnan [a], Chinmay Hegde [b], Soumik Sarkar [a], Baskar Ganapathysubramanian [a,*]

[a] Department of Mechanical Engineering, Iowa State University, Ames, IA 50010, USA
[b] Tandon School of Engineering, New York University, Brooklyn, NY, 11201, USA

ARTICLE INFO

Keywords:
Structure–property mapping
Multi-fidelity data
Deep learning
Organic electronics
Photovoltaics

ABSTRACT

Machine learning approaches have been used with significant success in constructing, curating, and exploring relationships between microstructure and property. However, one major limitation of these approaches is the need for a significant amount of training data consisting of microstructure–property pairs. Getting property values associated with a specific microstructure typically requires deploying a detailed physics simulator which becomes resource-intensive. While using a low(er) fidelity property quantifier can offset the cost of creating the training dataset, there is a trade-off in terms of accuracy/fidelity of the estimated property. Here, we leverage the availability of low- and high- fidelity property simulators to construct a multi-fidelity mapping from microstructure to property using deep convolutional neural networks. Starting with a large dataset of morphologies representing the active layer of organic photovoltaic devices, we assimilate data from a rapid graph-based low-fidelity characterization of the morphology with limited data from a high fidelity excitonic drift-diffusion detailed physics simulator. We show that our method provides significant computational savings while maintaining competitive performance. This work can be easily extended to other applications, and we envision it as a basis for accelerated material quantification and discovery.

1. Introduction

A pressing challenge in the design of new materials is the identification of microstructure–property linkages that can lead to microstructure-sensitive design. Numerous studies in literature address constructing fast structure–property predictions as a key step to identifying interesting microstructures [1,2]. An Edisonian approach typically involves screening through a large pool of computationally simulated candidate microstructures and performing an exhaustive evaluation of all microstructures. But screening such an expansive design space is computationally inefficient and not scalable, especially when using detailed physics simulators that are expensive to evaluate. With the recent advent of machine learning (ML) techniques, fast and accurate property predictions for multiple properties of interest have been demonstrated to be possible [3–6]. Such ML-based fast predictions can aid the construction and deployment of structure–property relationships in materials. However, a major bottleneck is that construction of these ML-based models require the availability of large amounts of training data, the gathering of which is again resource intensive.

On the other hand, for many materials design applications that involve mapping microstructure to property, there often exist a *spectrum* of physics simulators spanning from low-fidelity (but cheap to compute) to high-fidelity (but expensive to compute). Therefore, there is considerable impetus towards developing ML-based microstructure–property models that can assimilate such – unbalanced amounts of – multi-fidelity information.

Recent work in other engineering disciplines suggest such multi-fidelity assimilation can be highly successful. For example, multi-fidelity approaches have been used to construct surrogate models to investigate honeycomb crash-worthiness [7], for investigating the trade-off between accuracy and computational effort of calibrating structural dynamics computational models [8], and for seismic simulation of a virtual city with multi-fidelity building models [9]. Additionally, multi-fidelity information has also been shown to improve the efficiency of computationally expensive global optimization problems [10]. More recently, multi-fidelity approaches have also been applied to enhance the resolution of latent structural features in materials graphs [11] and to fuse atomistic computational model predictions across multiple levels of fidelity [6].

These studies demonstrate that multi-fidelity information assimilation could be beneficial from various aspects such as better prediction accuracy, faster computational speed, and better data efficiency. An open direction is the principled development of ML approaches – which are typically data hungry – that can assimilate multi-fidelity data,

* Corresponding author.
E-mail address: baskarg@iastate.edu (B. Ganapathysubramanian).

https://doi.org/10.1016/j.commatsci.2022.111599
Received 21 February 2022; Received in revised form 11 June 2022; Accepted 13 June 2022
Available online 22 July 2022

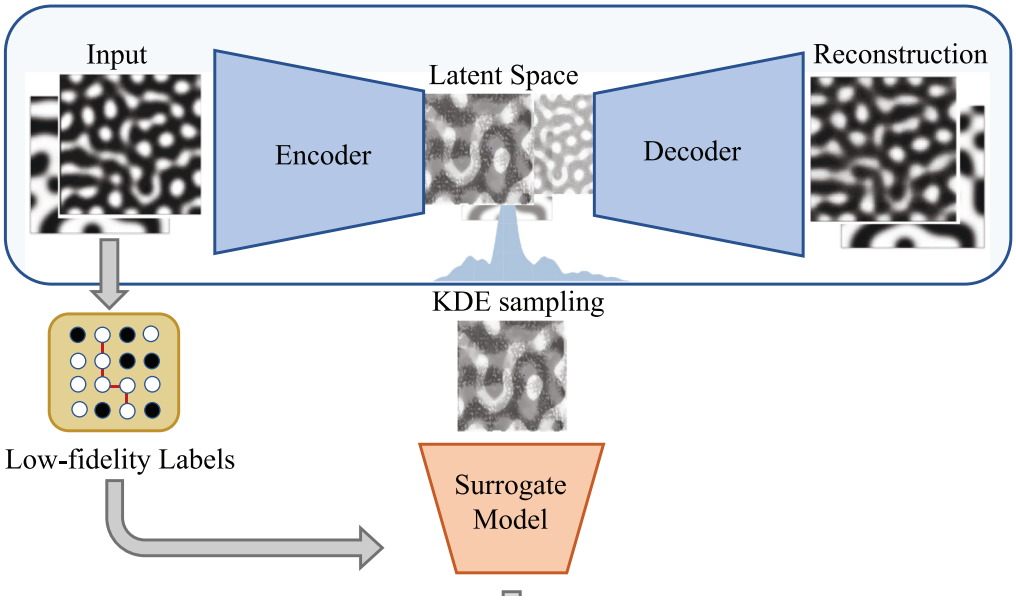


Fig. 1. Overview of multi-fidelity framework for OPV microstructure property prediction. The framework consists of an autoencoder, which learns a compressed latent representation of high-dimensional data. A neural-network based surrogate model is trained to predict full-fidelity properties based on the latent representation and low-fidelity labels generated by a graph-based black-box algorithm.

thus circumventing the high cost of creating a large training dataset. Such methods can potentially deliver high prediction accuracy, fast computational speed, and good transferability between models [12].

In this work, we propose an ML-based framework that incorporates the paradigm of multi-fidelity assimilation to obtain a data-driven structure–property prediction model. Our model is significantly more data-efficient than the current state-of-the-art while maintaining competitive predictive performance. The core idea of our framework relies on first learning a latent representation of high-dimensional microstructure images, and then assimilating the latent representations with computationally cheap, low-fidelity labels to predict full-fidelity properties. We demonstrate the efficacy of our framework on a structure–property prediction problem for organic photovoltaics. We utilize the framework to predict two microstructure-sensitive properties – the short circuit current J_{sc}, and the fill factor ff – that determine the performance of a photovoltaic device. We show that accurate ML models can be built using *nearly 2 orders of magnitude fewer* full-fidelity simulations, therefore adding weight to the promise of multi-fidelity information assimilation for practical materials design applications.

2. Methods

Our problem definition is as follows: Given a microstructure representing the active layer of an organic photovoltaic device, we seek to construct fast and accurate models that can predict the photovoltaic performance of that microstructure. We consider two-phase material systems, with the microstructure represented as a gray scale image, I. The photovoltaic performance is characterized by two properties – the short circuit current J_{sc}, and the fill factor, ff. We utilize a large dataset of morphologies containing over 65,000 samples. The photovoltaic performance of these morphologies can be evaluated in two ways – (a) a detailed physics photophysics simulator that provides accurate performance characterization, and (b) a low fidelity mechanistic model that can approximately predict performance.

We begin with an overview of the multi-fidelity framework for predicting high fidelity properties of a microstructure used for OPV applications in Section 2.1. We detail the dataset described above and the high/low fidelity property simulators in Section 2.2. We wrap up this section by detailing some of the network architecture choices we explore in Section 2.4

2.1. Overview of multi-fidelity framework

As illustrated in Fig. 1, our framework consists of an auto-encoder and a surrogate model, with both models represented using deep neural networks. To train this multi-fidelity model, we use a two-step training scheme as follows: First, we train the autoencoder to reconstruct images of the microstructures using an encoder–decoder architecture. An autoencoder compresses the input data into a low dimensional latent representation. The associated decoder allows recovery of the original image from this latent representation. The availability of a low dimensional representation allows creating more accurate mappings, given a finite amount of data.

The inputs to the encoder are 128 × 128 grayscale images, and the encoder network consists of convolutional and pooling layers which successively down-sample the 128 × 128 images into a compressed latent representation, denoted as X, with a dimension of 16 × 16. This represents a 64 fold reduction in dimensionality. The decoder network is a symmetric copy of the encoder, which then up-samples the latent space representation, X to reconstruct back the microstructure, $\tilde{I}$. The entire autoencoder was trained using a reconstruction loss shown in Eq. (1)

$$L_{recon} = 1/N \sum_{i=1}^{N} \langle I_i - \tilde{I}_i \rangle^2 \tag{1}$$

where N denotes the average over the samples (here the training batch-size), I_i the ith microstructure image with $\tilde{I}_i$ the corresponding reconstruction, and the difference is taken pixel-by-pixel.

Once the autoencoder is trained, we use a kernel density estimation (KDE) to non-parametrically represent the latent space. A KDE representation of the data distribution gives a straightforward way to diversely sample a tiny percentage of the data in the latent space (e.g., 1% of the entire training dataset) and access their corresponding high dimensional microstructure images. Sampling from the KDE ensures that the samples we select are diverse and representative of the entire set, which is especially important if the set is unbalanced and a large percentage of microstructures are biased towards specific properties. In order to quantify and compare diversity during sampling, we looked at the similarity of the datapoints in both the real and PCA space. While diversity in real space is preserved by using KDE (i.e., the

density estimate of the complete dataset and the sub-sampled dataset is the same in every dimension), we further quantified the diversity in PCA space. For this, we calculated the cosine similarity of first and second component of 100% training data PCA and 1% training data PCA, and the similarity is 87.02% and 87.18% respectively. For this smaller, diverse microstructure dataset, we utilize the high fidelity physics simulator to compute the photovoltaic properties.

We also generate low-fidelity descriptors, G, for the complete set of microstructures, which are computationally cheap to compute. Next, using the low-fidelity labels, G and the latent space representation, X as input, we train the surrogate model to predict the properties of the microstructure material in a supervised fashion using just a small fraction of full-fidelity labels. The loss function that the surrogate model is optimized over is a weighted mean square error (MSE) loss:

$$MSE = \frac{\lambda}{k}\sum_{i=1}^{N}(P_{true} - P_{predict})^2 \tag{2}$$

where λ denotes a heuristic parameter, k is the number of full-fidelity training labels, P_{true} is the ground truth full-fidelity labels and $P_{predict}$ is the predicted full-fidelity label. In our experiments, we set λ to 0.1 when training to predict J_{sc} and λ to 0.05 when training the model to predict ff. This λ value was set by observing the performance of the training from running multiple experiments.

During inference, we freeze the weights of the encoder and the surrogate model. Given a new microstructure image, a latent representation for this image is generated using the encoder. The low-fidelity labels for this image are generated using the low-fidelity simulator, and these two inputs (latent representation, and low fidelity labels) feed into the surrogate model to predict the desired properties, J_{sc} and ff.

As a baseline for comparing the efficacy of the proposed multi-fidelity framework, we consider a full-fidelity model. This full-fidelity model consists of the autoencoder and surrogate model, but predicts the properties solely from the latent space representation, *without* information from the low-fidelity labels.

The multi-fidelity model uses both the latent space representation with the low-fidelity labels as shown in Fig. 1. As such, both full-fidelity and multi-fidelity leverage the same autoencoder, and the only difference lies with the input to the surrogate model: in the case of the multi-fidelity model, the low-fidelity labels are concatenated with the flattened latent space vector; on the other hand, the input of full-fidelity surrogate models is simply the flattened latent space vector.

2.2. Data sets and property simulators

The data generation for this work consists of three distinct parts:

1. Creating the large dataset of morphology images
2. Low fidelity morphology quantification
3. Full fidelity morphology quantification

Morphology generation:. We have curated and open-sourced a large dataset of microstructure images arising from solving the Cahn–Hilliard equation with varying initial conditions.[1] The Cahn–Hilliard equation [13] describes phase separation occurring in a binary mixture, and has been shown to be a good representation of morphology evolution during fabrication of organic blend thin films [14–16] that are the typical active layer in OPV's. It tracks the evolution of local volume fraction of each phase, in the presence of spatial gradients in chemical potential of the system. Hence, in the time evolution process, one first observes an initial rapid separation of the well-mixed system into its constituent phases, followed by slow coarsening of the respective domains. Image data arising from the simulations provide a rich dataset for design of microstructures. Specifically, the morphologies obtained through simulation are visually similar to the morphologies in real active layer of a organic photovoltaic cells [17].

The Cahn–Hilliard equation models the spatio-temporal evolution of the local volume fraction ϕ of a two-component system:

$$\frac{\partial \phi}{\partial t} = \nabla\left(M\nabla\left(\frac{\partial f}{\partial \phi} - \epsilon^2\nabla^2\phi\right)\right) \tag{3}$$

where $f(\phi)$ is the local free energy density of the system, given by a Flory–Huggins formulation:

$$f(\phi) = \phi ln(\phi) + (1-\phi)ln(1-\phi) + \chi\phi(1-\phi) \tag{4}$$

χ represents the degree of interaction between the materials, and ϵ^2 is the energy of forming an interface between the two materials and is related to the interfacial thickness. M is the mobility of the components and is related to the diffusivity of each component. We use a well validated computational framework [2,18–21] to create this dataset. We perform multiple simulations with different blend ratios and different values of χ parameter. Specifically, we used 11 equispaced blend ratios from 0.50 to 0.63 and 11 equi-spaced χ values from 2.2 to 4.0. These morphologies were then thresholded by the initial blend ratio to get a binary image. The dataset was augmented by rotation, flipping and color inversion to get a large pool of binary morphology images to train on, resulting in a dataset with approximately 67k images.

2.3. Full-fidelity and low-fidelity simulators

In the context of our problem, we are interested in estimating the short circuit current density, J_{sc} and the fill-factor, ff of a given microstructure morphology. J_{sc} measures the current magnitude passing through the active layer morphology when the voltage, V across the device is zero. ff is defined as the ratio of maximum power to the maximum power possible (which is defined as the product of the J_{sc} with the open circuit voltage V_{oc}).

Full-fidelity simulator: Together, these two properties can provide a measure of the power conversion efficiency of a given morphology. One can calculate these two properties by solving a morphology aware (i.e. spatially heterogeneous) photophysics device model. The photophysics model is described by the steady state *excitonic drift diffusion (XDD) equations*. The XDD equations are a set of four tightly coupled partial differential equations that model the optoelectronic physics of energy harvesting in organic photovoltaic devices. We deploy a validated, in-house software that uses a finite element based solution strategy for solving the photophysics device model. We refer the interested reader to previous papers for a detailed description of the solver [22–24].

However, calculating these full-fidelity properties are computationally expensive. For example, computing the J_{sc} and ff descriptors of a single morphology can take up to 72 min [25] on a single node of a high performance computing system. Ideally, we seek to limit the number of morphologies for which we deploy this full-fidelity simulator. However, to test the effectiveness of the proposed multi-fidelity approach, we were able to calculate the full-fidelity simulations for approximately 34,000 microstructures. This represents a significant computational effort, and is usually infeasible to repeat for other problems.

Low-fidelity simulator: To circumvent the computational complexity of solving the XDD equations, previous works have developed a low-fidelity graph-based approach to construct reduced order representation of the photovoltaic performance [1,26–28]. This approach converts the microstructure image into an equivalent graph with binary (black and white) nodes and calculates various properties of the graph, which are highly correlated with a various photophysics phenomena occurring in the active layer. Some of the graph based descriptors that have been shown to be well correlated with performance (as discussed in [29]) include: (i) light absorption (but donor region) represented by the fraction of black (i.e. donor) nodes, (ii) exciton diffusion represented by the interfacial area between the black and white nodes, and

[1] The dataset is available under the Creative Commons 4.0 license and can be downloaded at: https://zenodo.org/record/2580293#.XJGDEC2ZPox.

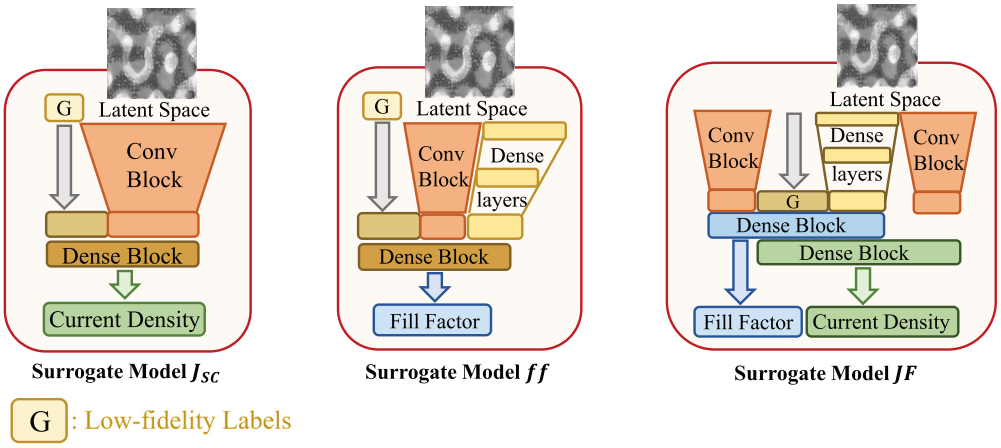


Fig. 2. Illustration of the different architecture of surrogate models investigated for OPV property prediction. Model J_{sc} and model ff were trained to only predict their respective properties individually while model JF was trained to predict both J_sc and ff simultaneously. Results revealed that models J_{sc} and ff performs marginally better than model JF.

(iii) charge transport, represented by statistics of graph shortest paths in the graph. It has been shown that the product of these descriptors is highly correlated with J_{sc} [1,2]. While fast and computationally cheap, these descriptors do not provide an accurate prediction of J_{sc} and ff.

2.4. Exploring different network architectures for constructing the surrogate model

We experiment with three different architectures to build both full-fidelity and multi-fidelity surrogate models: (i) model J_{sc} which only predicts J_{sc} values, (ii) model ff, which only predict ff values and (iii) model JF which jointly predicts both J_{sc} and ff values. Given that both J_{sc} and ff represent different photophysics phenomena, it is reasonable to expect that the morphology features required to predict each property may not be identical (e.g., predicting ff may require the model to learn more features than predicting J_{sc}).

To explore this hypothesis, we designed three distinct surrogate models to determine which approach is more suited for predicting the properties with higher accuracies. Surrogate model J_{sc} consists of a convolutional block, a concatenation step with the low-fidelity descriptors followed by a dense block. In surrogate model ff, we use parallel convolutional and dense blocks followed by a concatenation step with the low fidelity output before passing the logits through a final dense block. Lastly, the surrogate model JF consists of two separate but parallel convolution blocks and a dense block. The outputs of these network blocks are concatenated with the low fidelity label before going through separate dense blocks for J_{sc} and ff predictions. The architectures of these three surrogate models are graphically visualized in Fig. 2. The full-fidelity surrogate models also use the same architectures described above, except that the logits from the convolutional and dense blocks were not concatenated with the low-fidelity labels.

3. Results

3.1. Accuracy and robustness of autoencoder

We first present the results of training the autoencoder, as the quality of the latent representation depends significantly on the autoencoder's ability to reconstruct the microstructure images. Using the dataset detailed in Section 2.2 (approximately 67k images), we randomly split 80% of the data in the training set and set the remainder 20% as the testing set. The autoencoder model was trained for a total of 450 epochs, although the model visibly converged after 300 epochs, as seen in Fig. 3a. The average reconstruction loss for random 1000 images from training images set and testing images set are 1.72E−3 and 1.71E−3 respectively. This indicates that the encoder–decoder pair constituting the autoencoder have learnt a good bijective mapping to a low-dimensional latent representation.

Figs. 4(a) and 4(b) visualizes samples of the microstructure from the training set and testing set, respectively. The top rows in each figure are the actual samples from the dataset, while the bottom rows show the reconstructed images (i.e. the result of the decoder operating on the latent representation). As can be seen in the figure, for both randomly selected training and testing microstructures, the autoencoder is capable of compressing the microstructure images into a latent representation and reconstructing them with no significant visual difference. This observation of the autoencoder's generalizability from the training set to the testing set can also be concluded from the vanishingly small reconstruction loss of the autoencoder on the 1000 images randomly sampled from both the training and testing set. To further demonstrate the robustness of the autoencoder model on out-of-distribution data, we constructed additional images with saw-tooth-like structures, shown in the top row of Fig. 4(c). These images are significantly different from any samples of the microstructure dataset and consists of features such as sharp corners, straight and orthogonal lines, which are typically not observed in the microstructures data. As such, these saw-tooth morphologies represent data that lies outside of the training distribution. To illustrate and validate that these saw-tooth-like structures are different from the training microstructure dataset, we first applied Principal component analysis (PCA) on the training images set and the sawtooth morphology, then project the first 200 PCA components onto a two-dimensional plane via t-Distributed Stochastic Neighbor Embedding (t-SNE) [30]. The t-SNE projection is shown in Fig. 3b and it can be seen that the saw tooth images are indeed lying on the fringes of the microstructure dataset, indicating that they are statistically out-of-distribution samples.

Fig. 4(c) shows how the autoencoder performs on these out-of-distribution samples. As observed in the bottom row of Fig. 4(c), the autoencoder can reconstruct these images although it has never been trained on such images before. The mean reconstruction MSE loss of these five saw-tooth-like images was as 7.18E−3. This suggests that the autoencoder has learnt an accurate transformation of the high-dimensional microstructure images into a compressed latent representation. Furthermore, this

3.2. Comparing multi-fidelity with full-fidelity models

Next, we present the results from comparing the performance of our multi-fidelity framework with the full-fidelity framework for predicting the J_{sc} and ff values from the morphology images. Using the three different architectures described in Section 2.4, we trained the surrogate

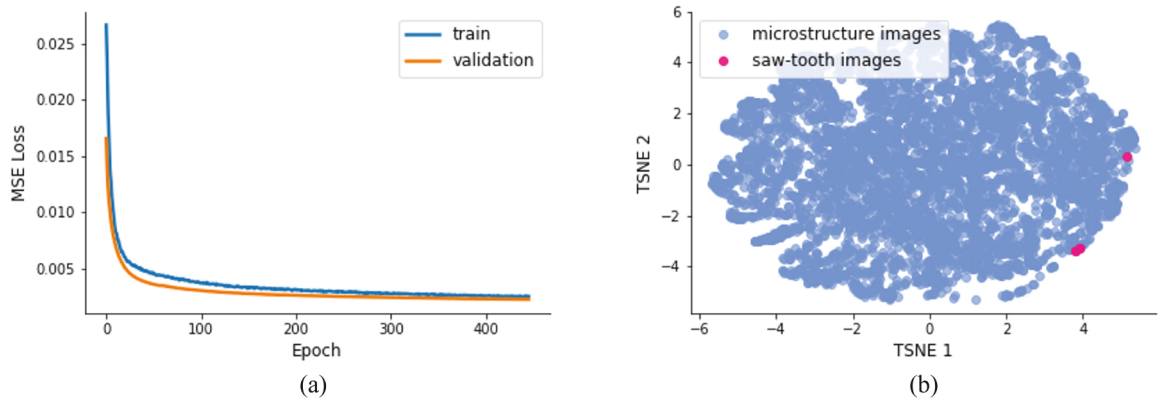
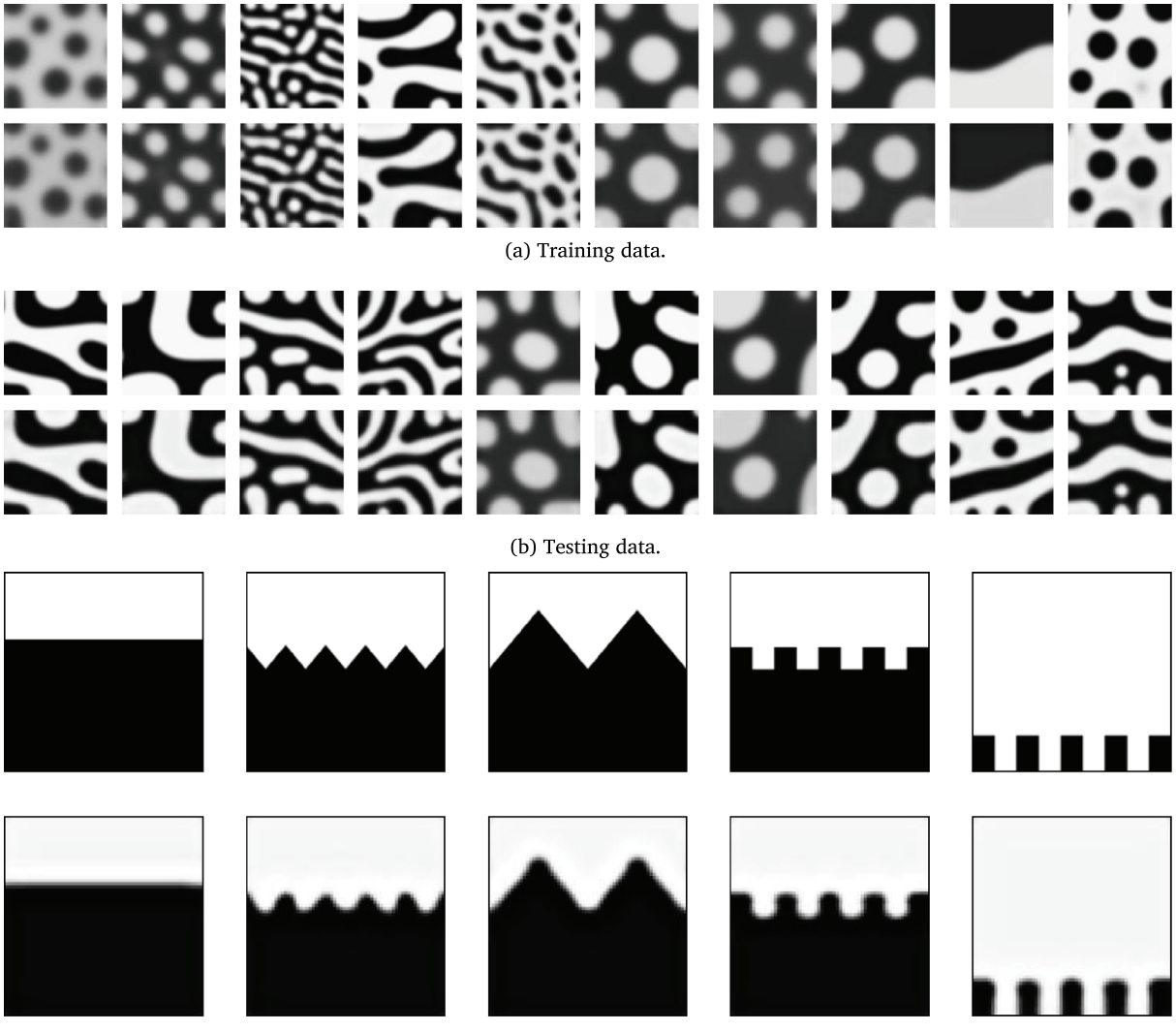


Fig. 3. Loss trends and visualization of out-of-distribution data (a) Training and validation loss-history of the autoencoder. The loss values plateaus after 450 epochs (b) t-SNE visualization of the first 200 principal components of the microstructure images. Images from the training dataset are shown in blue and images of the saw-tooth microstructures are shown in magenta. Observe that the saw-tooth like images lies on the edge of the entire training data distribution.



(a) Training data.

(b) Testing data.

(c) Outlier data.

Fig. 4. Performance of the autoencoder for reconstruction. We demonstrate the working of the autoencoder with (a) Training data, (b) Testing data and (c) Outliers. The outliers are morphologies that are of a completely different nature than the training/testing data.

models to predict J_{sc} and ff from the latent space representations (and low-fidelity labels for the multi-fidelity model) in a supervised fashion using different amounts of labels, from 1% to 100% of the training dataset and evaluated the performance of these models on the test set.

Note that 100% of the training set here refers to 34k data points, which is the full dataset on which we were able to perform the full-fidelity simulations.

Fig. 5 shows the R^2 values of the predicted properties with the ground truth properties for different percentages of training data used. As shown in both figures, the multi-fidelity models (shown in blue lines) outperform the full-fidelity models (shown in green lines) for all percentages of training data used, regardless of the model architectures. Additionally, we noticed that multi-fidelity models also tend to converge faster and perform better than the full-fidelity model. For example, in Fig. 5b, the performance of model J_{sc} trained on multi-fidelity data stopped improving significantly with 15% of the data while model J_{sc} trained on full-fidelity data only converges approximately with around 50% of the data. Meanwhile, for ff predictions, the multi-fidelity model converges with 50% of the training data while the full-fidelity model shows no signs of converging even with 100% of the training data, as seen in Fig. 5a. In addition, note that the performance of the full-fidelity model ff never truly reached the performance of the multi-fidelity model ff, even when provided with the entire training set. The improved accuracy of the multi-fidelity surrogate models over full-fidelity models is exciting, especially in the low training data regime. These observations imply that the low-fidelity descriptors do contain useful information that can be leveraged to boost the model's performance when training with limited data and for more accurate property predictions. Finally, we note the difference in predictive performance – of both full-fidelity and multi-fidelity models – across the two properties, J_{sc} and ff. We attribute this to the more complex relationship between morphology and ff that is only partially captured by the features provided. In general, Jsc depends only on the morphology. On the other hand, FF is a more complex property (which is a summary statistic of the current voltage plot), and depends not only on the morphology but also on material properties, and the shape of the current–voltage plot.

To compare the effects of the different surrogate network architecture on prediction accuracy, we observe that the performance of the models trained to predict a single property, shown in solid lines (model J_{sc} or model ff) are marginally better than the performance of the models trained to predict both properties simultaneously, shown in dotted lines (model JF) for both multi-fidelity and full-fidelity. This tells us that the choice of model architecture does affect the performance to a certain extent. Nevertheless, the observation that using low-fidelity descriptors in tandem with the latent space representation improves the model's prediction accuracy is still valid. Hence, in the following sections, we focus our experimentation on models J_{sc} and ff. As a further illustrative comparison between the performance of multi-fidelity and full-fidelity models, the predicted microstructure properties from the models trained with only 1% of the full-fidelity data and their corresponding ground-truth values are plotted in the rows of Fig. 6. Additionally, the error distributions for the predicted properties are plotted in the bottom row of these figures. As seen in Fig. 6a and b for the predictions of J_{sc}, the usage of low-fidelity data increased the R^2 value by 12.40%. Meanwhile, for predictions of ff, the R^2 value improved by 36.48% when we incorporated the low-fidelity information into the surrogate model's input. Additional information and discussion on model JF are provided in the Appendix.

3.3. Quantifying the efficiency gain of multi-fidelity model

The improved performance of the multi-fidelity models (using the latent space representation and low-fidelity labels supervised by full-fidelity labels) implies that we can decrease the number of full-fidelity training labels and still achieve competitive results with the help of multi-fidelity data. In other words, we can improve the data efficiency of training a predictive model via multi-fidelity models.

To objectively quantify the amount of data efficiency we gained by leveraging the low-fidelity labels, we performed further analysis on the results obtained from the experiments presented in the previous section. Using the full-fidelity model's test results trained on 100% of the dataset as a benchmark, we analyze the multi-fidelity model's test results trained on 1% of the data and compare their performance in terms of the R^2 and error distributions. We then progressively increased the size of the training data of the multi-fidelity model until the error distributions and R^2 values matched the performance of the full-fidelity model. We find that for the prediction of J_{sc}, only 3% of the entire training data is needed for the multi-fidelity model (model J_{sc}) to be competitive with the full-fidelity model trained on 100% of the dataset. For predictions of ff, model ff requires only 25% of training data required to reach the same performance as the full-fidelity data. In other words, using the multi-fidelity model yields a data efficiency gain of 33X and 4X for model J_{sc} and model ff, respectively. The results of this analysis are also illustrated in Fig. 7 for predictions of J_{sc} and ff. In the figures, the top row shows the prediction versus the ground truth values of the property, and the bottom rows show the distribution of prediction errors. The first and third columns represent the performances of the full-fidelity models trained on the entire training set, and the second and fourth columns represent the multi-fidelity models trained on a smaller percentage of data. As observed from Fig. 7, the R^2 and error distribution of the multi-fidelity model is very close to the full-fidelity model.

4. Discussions and future work

In this section, we further elaborate on the benefits of our two-step training approach for the multi-fidelity framework and its implications. Note that in the microstructure property-prediction problem we are trying to address, the main computational constraints come from the fact that computing full-fidelity properties of the microstructures are computationally prohibitive. As such, it is only feasible to generate a small number of such labels. In our framework, the autoencoder is trained in an unsupervised manner since no labels are required to train the model to reconstruct the image. Hence, we can ensure that the latent representation learned by the autoencoder is well-trained and generalizable by training the autoencoder on a large set of training data since obtaining images of the microstructures is not a bottleneck. However, to predict the full-fidelity properties solely based on the latent space representation may be challenging due to various aspects such as the limited capacity of the latent representation and over-fitting to the training dataset. Combining additional information such as the low-fidelity labels with the latent representation allows us to train a surrogate model with high prediction accuracy while only using a small amount of computationally expensive full-fidelity labels. Therefore, by training the autoencoder, then the surrogate model in a two-step approach, we are essentially decoupling the cheap microstructure generation process from the expensive descriptor generation process. Moreover, given that the difficulty of obtaining various kinds of data is different, the cost of getting valid simulation labels might also be different. Thus, for better data efficiency, a framework that can use more data that is computationally cheap and accessible while assimilating fewer data that are expensive/hard to generate is essential. Therefore, the modularity of this framework is highly beneficial as it allows us to train the autoencoder and the surrogate models with different sizes of training datasets. In the cases we presented here, we have the morphology data being the larger dataset (68k images), and the properties J_{sc} and ff being the smaller dataset (34k labels). Furthermore, by training the autoencoder separately, we end up with a model which can be used for other downstream applications, such as for design exploration or for property optimization.

Next, we briefly highlight another advantage of our multi-fidelity framework approach. From the results presented, we have shown that the multi-fidelity models consistently performed better than full-fidelity models in predicting the microstructure's property while being significantly more data-efficient. However, there is no major increase in

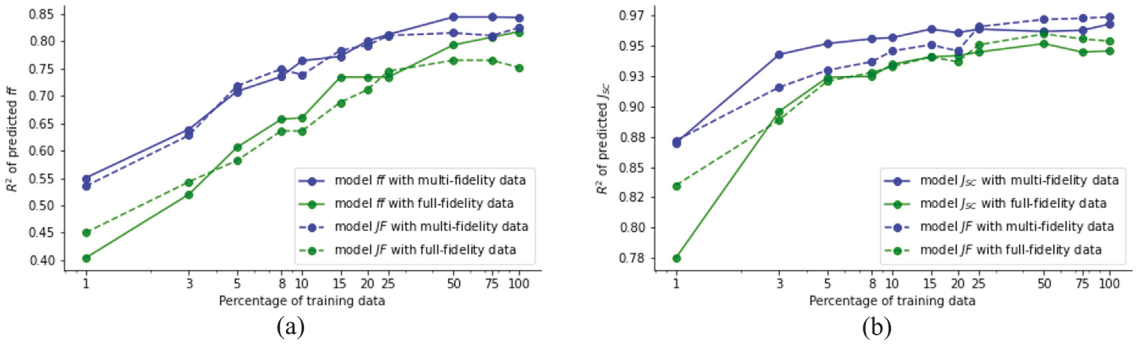


Fig. 5. Comparison of prediction accuracy between full-fidelity and multi-fidelity models for different percentage of training data. (a) Prediction accuracies of full-fidelity and multi-fidelity models for prediction of ff, (b) Prediction accuracies of full-fidelity and multi-fidelity models for prediction of J_{sc}. Observe that multi-fidelity models consistently outperforms full-fidelity models and models trained to predict single property (model J_sc and model ff) slightly outperforms model that is trained to predict both properties simultaneously (model JF).

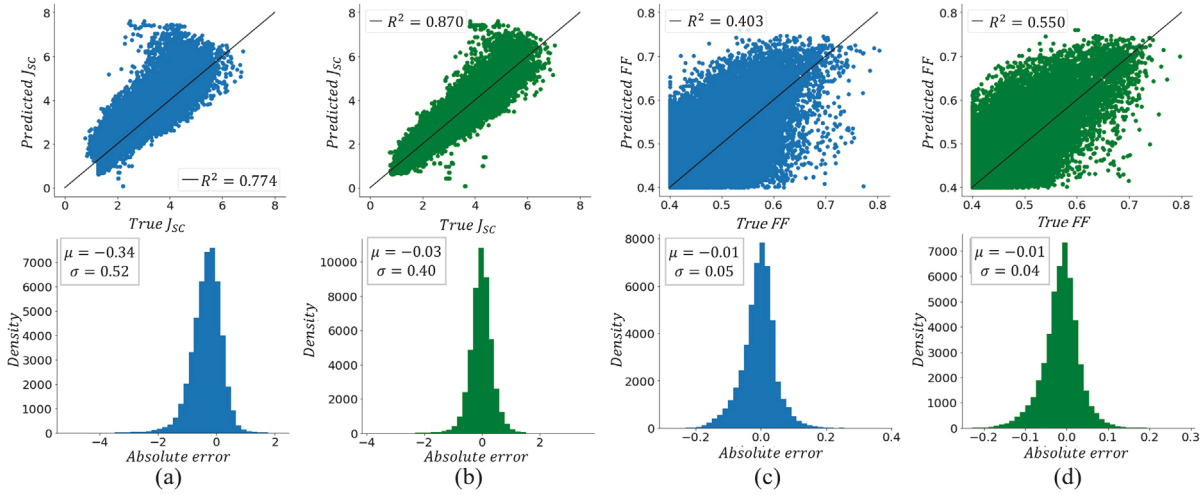


Fig. 6. Comparison of prediction accuracy and error distribution between full-fidelity and multi-fidelity models when trained on 1% of training data. Figures on the top row illustrates the predictions versus ground truth values in the testing set for (a) model J_{sc} trained with only full-fidelity data (b) model J_{sc} trained with multi-fidelity data (c) model ff trained with only full-fidelity data (d) model ff trained with multi-fidelity data. The figures on the bottom row illustrates the distribution of prediction errors for each model respectively. With only 1% of training data, leveraging additional information in the form of low-fidelity labels provides the multi-fidelity model significant improvements in accuracy.

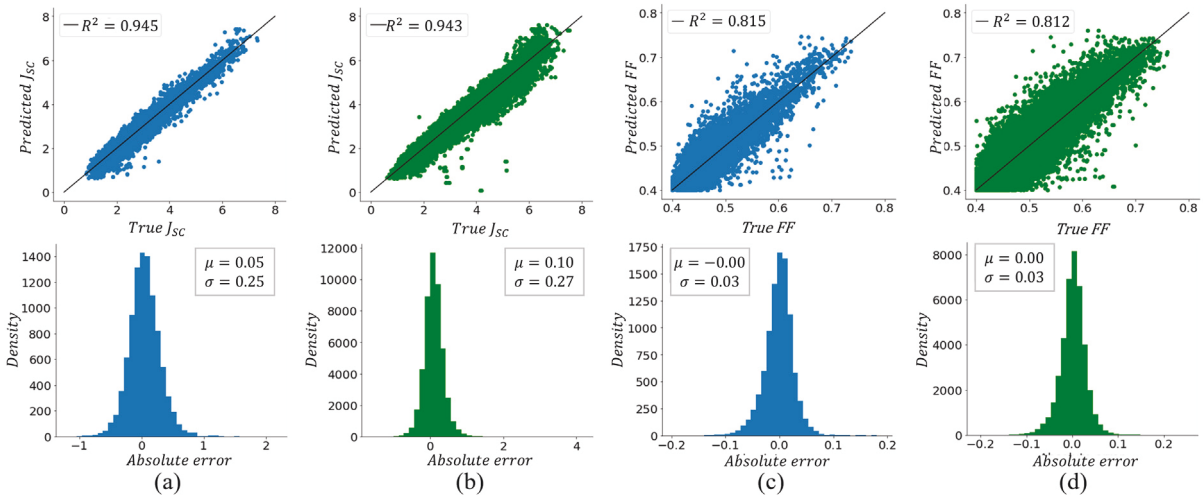


Fig. 7. Matching the performance of full-fidelity model trained on entire dataset with multi-fidelity model trained on partial dataset. (a) The performance of the full-fidelity model J_{sc} trained on 100% data is matched by (b) multi-fidelity model J_{sc} trained only on 3% of data. (c) The performance of the full-fidelity model ff trained on 100% data is matched by (d) multi-fidelity model ff trained only on 25% of data.

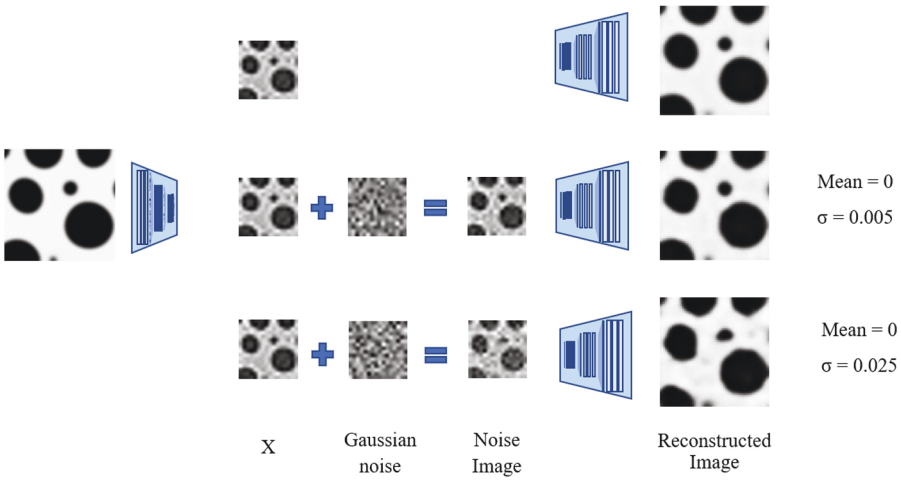


Fig. A.8. Morphology reconstruction by the autoencoder with Gaussian noise added to the latent space: Observe that adding noise to the latent space results in reconstructed images with noisier features, although the main structural features remain intact.

Table 1
Number of trainable parameters in the three proposed surrogate model architectures. Note that the number of trainable parameters in the multi-fidelity models is roughly similar to the number of trainable parameters in the full-fidelity models. Corresponding architectures are illustrated in 2. Here, we have used relatively shallow surrogate models since the input to the surrogate models are the low-dimensional latent vectors that are compressed from the original morphology images.

Model	Number of trainable parameters	
	Full-fidelity	Multi-fidelity
Model J_{sc}	251	254
Model ff	6174	6182
Model JF	6298	6358

model complexity by employing the multi-fidelity model. Specifically, the multi-fidelity models are only 0.13%, 1.2%, and 0.95% larger than the full-fidelity models for model J_{sc}, model ff and model JF respectively. Additionally, we also observed no noticeable increase in training time for multi-fidelity models compared to full-fidelity models. This is because the network architecture used in the full and multi-fidelity models are mostly similar, with the input layer of the multi-fidelity surrogate model having additional parameters to account for the additional low-fidelity label input. As a reference, specific details on the numbers of trainable parameters in each model are shown in Table 1. In summary, the multi-fidelity approach provides a huge benefit of better performance and data efficiency but with no major cost of increased model complexity.

While there are several different exciting avenues to pursue in terms of future direction, we focused this discussion on further improving the data efficiency of our proposed multi-fidelity approach. Specifically, one future extension of this work is to incorporate active learning paradigms within our framework. In the multi-fidelity framework presented, we have used a kernal density representation followed by sampling to sample a small percent of the data representing the latent space density estimated by the KDE. However, such a sampling scheme might not be the most efficient as the representative samples of the latent space might not be samples that are representative of the J_{sc} and ff space. This insight has also led us to using a weighted MSE loss instead of a regular MSE loss as discussed in Section 2.1. Instead, it might be more beneficial to sample data points that are representative of latent space, which are also outliers in the J_{sc} and ff domains to maximize the information gain from these samples. Thus, future work will include an active learning framework where we first train the multi-fidelity model with minimal data (e.g., 1%) and make predictions on the initial batch of data. Based on the predictions, we can select the microstructures with outliers descriptors and sample similar microstructures in the latent space. These newly sampled microstructures are then added to the training set, and the multi-fidelity surrogate model is re-trained again. With this framework, we gradually increase the training dataset size until the performance of the surrogate model is satisfactory [31].

5. Conclusion

This work demonstrates a working example of how deep neural networks can be used to assimilate multi-fidelity information to reduce the need for high-cost full-fidelity quantification. More specifically, in the context of OPVs, we showed that multi-fidelity networks save significant computational costs by being more data-efficient. The multi-fidelity network works by first transforming the morphology manifold (images of microstructure) into a low dimensional continuous latent representation using a vanilla autoencoder. Simultaneously, we also compute several low-fidelity metrics of the microstructure images that are of interest to the OPV community, such as domain sizes and connectivities via a fast graph-based method. Next, the latent representation is combined with low-fidelity metrics to predict full-fidelity metrics such as short circuit current (J_{sc}) and fill-factor (ff). We showed that this staggered multi-fidelity data assimilation technique helps reduce the high fidelity data requirement by at least four times while maintaining high prediction accuracy.

The work opens up several other challenges in multi-fidelity modeling. The presented autoencoder model might not fully capture the manifold of the training data (i.e., interpolation of the latent representation does not truly interpolate the image manifold) as compared to the interpolative capabilities that are inherent in other architectures, such as generative adversarial models [32,33]. Another aspect of the architecture is the ability to give interpretable latent representations. Such interpretability [4] is particularly needed for material science systems, where manufacturability of a particular morphology is severely limited by the existing processing techniques. Another avenue for research could be relating the latent representation on the manifold to processing conditions that intuitively creates an interpretable latent representation.

CRediT authorship contribution statement

Chih-Hsuan Yang: Methodology, Software, Validation, Writing – original draft, Visualization. **Balaji Sesha Sarath Pokuri:** Methodology, Software, Validation, Data curation, Writing – original draft. **Xian Yeow Lee:** Conceptualization, Methodology, Software, Writing

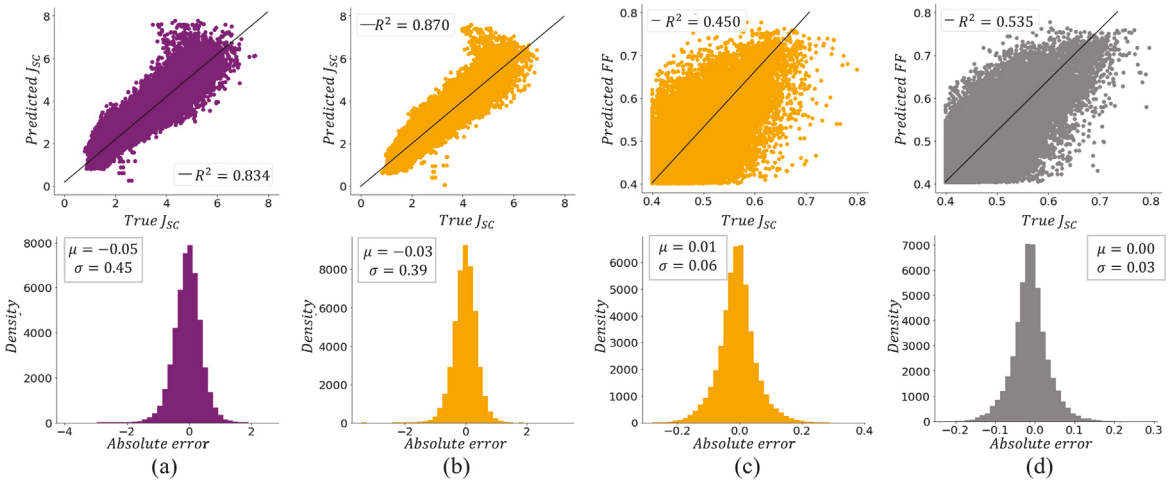


Fig. B.9. Comparison of prediction accuracy and error distribution between full-fidelity model JF and multi-fidelity model JF when trained on 1% of training data. (a) performance on predicting J_{sc} of model JF trained with only full-fidelity data (b) performance on predicting J_{sc} of model J_{sc} trained with multi-fidelity data (c) performance on predicting ff of model JF trained with only full-fidelity data (d) performance on predicting ff of model JF trained with multi-fidelity data. The figures on the bottom row illustrates the distribution of prediction errors for each model respectively. With only 1% of training data, leveraging additional information in the form of low-fidelity labels provides the multi-fidelity model significant improvements in accuracy.

– original draft. **Sangeeth Balakrishnan:** Investigation, Methodology. **Chinmay Hegde:** Conceptualization, Writing – review & editing, Supervision, Funding acquisition. **Soumik Sarkar:** Conceptualization, Writing – review & editing, Supervision, Funding acquisition, Project administration. **Baskar Ganapathysubramanian:** Conceptualization, Writing – original draft, Writing – review & editing, Supervision, Funding acquisition, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability statement

The raw data required to reproduce these findings are available to download from here [34]. High fidelity and low fidelity information will be made public/is available upon request from the corresponding authors.

Acknowledgments

This work was supported by the ARPA-E DIFFERENTIATE program, USA under grant DE-AR0001215. BG, C-HY, and BP were supported in part by DoD MURI, USA 6119-ISU-ONR-2453 and NSF 1906194. CH was supported in part by NSF, USA grants 2005804 and 1815101. Computing support from XSEDE, USA and Iowa State University, USA is gratefully acknowledged.

Appendix A. Architecture and robustness of autoencoder

The details of autoencoder architecture are introduced below. The autoencoder can be separated into two symmetric parts: the encoder and the decoder. As seen in Table A.2, the 128 × 128 × 1 active layer morphology image was passed through a convolution layer, max-pooling layer, and dropout layer to a 64 × 64 × 128 feature map, then continuously passed into another set of convolution layer, max-pooling layer, and dropout layer. After this, a 32 × 32 × 64 feature map was obtained. This feature map then went through a final set of convolution layers, max-pooling layer, and dropout layer to transform into a 16 × 16 × 32 feature map. The 16 × 16 × 1 latent space mentioned in the main article is gained after the 16 × 16 × 32 feature map was passed through a convolution layer. The decoder part of this autoencoder has a symmetric structure with the encoder part: it successively magnifies the latent space into 16 × 16 × 32 feature map, 32 × 32 × 64 data 64 × 64 × 128 feature map, then 128 × 128 × 1 image as the same structure of original active layer morphology image. The total trainable parameter number of this autoencoder is 187,364 and has been trained for 450 epochs.

Table A.2
Details of autoencoder architecture. The encoder encodes morphology images from 128×128 dimension to 16×16 dimensional latent space and with the decoder having the same symmetric structure as the encoder.

Layer no.	Input shape from the layer	
	Encoder	Decoder
1	128×128×1	16×16×1
2	64×64×128	16×16×32
3	32×32×64	32×32×64
4	16×16×32	64×64×128
Output	16×16×1	128×128×1

We provide additional anecdotal but empirical results here that demonstrate the robustness of the autoencoder in reconstructing the microstructures, even in the presence of noise (see Fig. A.8). Having a robustly trained autoencoder will increase our confidence in the autoencoder's ability to represent the high dimensional images in a lower-dimensional latent space, which is critical for the performance of the surrogate model downstream. Since the latent representation characterizes the essential low dimensional features of the high dimensional data, we analyze the effect of adding perturbations (specifically Gaussian noise) to the latent features on the reconstructed image. As observed in Fig. A.8, adding noise to the latent features does result in slightly more noisy features when the morphologies are reconstructed. Nevertheless, the main features that define the structure of the morphology remain intact. Future work will also focus on developing a model that is more robust towards noise while having interpolative capabilities like generative models, as discussed in the conclusion section of the main article.

Table B.3
Comparison of full- and multi-fidelity surrogate model's accuracy for different ratios of training data for model JF. Observe that as we drastically reduce the size of training data set, the performance of the full-fidelity surrogate degrades much more significantly than the multi-fidelity surrogate model.

Percent of data	Full-fidelity surrogate (R^2)		Multi-fidelity surrogate (R^2)	
	J_{sc}	ff	J_{sc}	ff
100	0.954	0.752	0.974	0.824
75	0.956	0.765	0.973	0.810
50	0.960	0.765	0.972	0.815
25	0.951	0.745	0.966	0.810
20	0.937	0.711	0.946	0.792
15	0.941	0.688	0.951	0.783
10	0.933	0.636	0.946	0.738
8	0.928	0.636	0.937	0.749
5	0.921	0.582	0.930	0.718
3	0.899	0.543	0.916	0.628
1	0.834	0.450	0.872	0.535

Appendix B. Additional results and discussion on surrogate model JF

Here, we compare the performance of model JF trained with full-fidelity and multi-fidelity data using only 1% of full-fidelity labels in Fig. B.9. From the figure, we verify that the conclusion that the usage of multi-fidelity labels increases the performance of the models in predicting both J_{sc} and ff is still valid. Additionally, the detailed results of the model trained with different percentages of full-fidelity labels are also tabulated in Table B.3. From the table, we can also observe that the model trained with multi-fidelity labels performs better than the model trained only with full-fidelity labels for all percentages of training data and for both predictions of J_{sc} and ff

References

[1] Olga Wodo, Srikanta Tirthapura, Sumit Chaudhary, Baskar Ganapathysubramanian, A graph-based formulation for computational characterization of bulk heterojunction morphology, Org. Electron. 13 (6) (2012) 1105–1113.

[2] Olga Wodo, Jaroslaw Zola, Balaji Sesha Sarath Pokuri, Pengfei Du, Baskar Ganapathysubramanian, Automated, high throughput exploration of process–structure–property relationships using the MapReduce paradigm, Mater. Discov. 1 (2015) 21–28.

[3] Keith T. Butler, Daniel W. Davies, Hugh Cartwright, Olexandr Isayev, Aron Walsh, Machine learning for molecular and materials science, Nature 559 (7715) (2018) 547–555.

[4] Balaji Sesha Sarath Pokuri, Sambuddha Ghosal, Apurva Kokate, Baskar Ganapathysubramnian, Soumik Sarkar, Interpretable deep learning for guided structure-property explorations in photovoltaics, 2018, arXiv preprint arXiv:1811.06067.

[5] Spencer Pfeifer, Balaji Sesha Sarath Pokuri, Pengfei Du, Baskar Ganapathysubramanian, Process optimization for microstructure-dependent properties in thin film organic electronics, Mater. Discov. 11 (2018) 6–13.

[6] Anh Tran, Julien Tranchida, Tim Wildey, Aidan P. Thompson, Multi-fidelity machine-learning with uncertainty quantification and Bayesian optimization for materials design: Application to ternary random alloys, J. Chem. Phys. 153 (7) (2020) 074705.

[7] Guangyong Sun, Guangyao Li, Michael Stone, Qing Li, A two-stage multi-fidelity optimization procedure for honeycomb-type cellular materials, Comput. Mater. Sci. 49 (2010) 500–511.

[8] Ghina N. Absi, Sankaran Mahadevan, Multi-fidelity approach to dynamics model calibration, Mech. Syst. Signal Process. 68–69 (2016) 189–206.

[9] Zhen Xu, Xinzheng Lu, Kincho H. Law, A computational framework for regional seismic simulation of buildings with multiple fidelity models, Adv. Eng. Softw. 99 (2016) 100–110.

[10] Bo Liu, Slawomir Koziel, Qingfu Zhang, A multi-fidelity surrogate-model-assisted evolutionary algorithm for computationally expensive optimization problems, J. Comput. Sci. 12 (2016) 28–37.

[11] Chi Chen, Yunxing Zuo, Weike Ye, Xiangguo Li, Shyue Ping Ong, Learning properties of ordered and disordered materials from multi-fidelity data, Nat. Comput. Sci. 1 (1) (2021) 46–53.

[12] Gisbert Schneider, Virtual screening: an endless staircase? Nat. Rev. Drug Discov. 9 (4) (2010) 273.

[13] John W. Cahn, John E. Hilliard, Free energy of a nonuniform system. I. Interfacial free energy, J. Chem. Phys. 28 (2) (1958) 258–267.

[14] Olga Wodo, Baskar Ganapathysubramanian, Modeling morphology evolution during solvent-based fabrication of organic solar cells, Comput. Mater. Sci. 55 (2012) 113–126.

[15] Olga Wodo, Baskar Ganapathysubramanian, How do evaporating thin films evolve? Unravelling phase-separation mechanisms during solvent-based fabrication of polymer blends, Appl. Phys. Lett. 105 (15) (2014) 153104.

[16] Kui Zhao, Olga Wodo, Dingding Ren, Hadayat Ullah Khan, Muhammad Rizwan Niazi, Hanlin Hu, Maged Abdelsamie, Ruipeng Li, Er Qiang Li, Liyang Yu, et al., Vertical phase separation in small molecule: polymer blend organic thin film transistors can be dynamically controlled, Adv. Funct. Mater. 26 (11) (2016) 1737–1746.

[17] Balaji Sesha Sarath Pokuri, Sambuddha Ghosal, Apurva Kokate, Soumik Sarkar, Baskar Ganapathysubramanian, Interpretable deep learning for guided microstructure-property explorations in photovoltaics, Npj Comput. Mater. 5 (1) (2019) 1–11.

[18] Olga Wodo, Baskar Ganapathysubramanian, Modeling morphology evolution during solvent-based fabrication of organic solar cells, Comput. Mater. Sci. 55 (2012) 113–126.

[19] Balaji Sesha Sarath Pokuri, Baskar Ganapathysubramanian, Morphology control in polymer blend fibers—a high throughput computing approach, Modelling Simulation Mater. Sci. Eng. 24 (6) (2016) 065012.

[20] Spencer Pfeifer, Balaji Sesha Sarath Pokuri, Olga Wodo, Baskar Ganapathysubramanian, Quantifying the effects of noise on early states of spinodal decomposition:: Cahn–Hilliard–Cook equation and energy-based metrics, in: Uncertainty Quantification in Multiscale Materials Modeling, Elsevier, 2020, pp. 301–327.

[21] Olga Wodo, Baskar Ganapathysubramanian, Computationally efficient solution to the Cahn–Hilliard equation: Adaptive implicit time schemes, mesh sensitivity analysis and the 3D isoperimetric problem, J. Comput. Phys. 230 (15) (2011) 6037–6060.

[22] Hari K. Kodali, Baskar Ganapathysubramanian, A computational framework to investigate charge transport in heterogeneous organic photovoltaic devices, Comput. Methods Appl. Mech. Engrg. 247 (2012) 113–129.

[23] Hari K. Kodali, Baskar Ganapathysubramanian, Sensitivity analysis of current generation in organic solar cells—comparing bilayer, sawtooth, and bulk heterojunction morphologies, Sol. Energy Mater. Sol. Cells 111 (2013) 66–73.

[24] Olga Wodo, Jaroslaw Zola, Balaji Sesha Sarath Pokuri, Pengfei Du, Baskar Ganapathysubramanian, Automated, high throughput exploration of process–structure–property relationships using the mapreduce paradigm, Mater. Discov. 1 (2015) 21–28.

[25] Xian Yeow Lee, Joshua R Waite, Chih-Hsuan Yang, Balaji Sesha Sarath Pokuri, Ameya Joshi, Aditya Balu, Chinmay Hegde, Baskar Ganapathysubramanian, Soumik Sarkar, Fast inverse design of microstructures via generative invariance networks, Nat. Comput. Sci. 1 (3) (2021) 229–238.

[26] Balaji Sesha Sarath Pokuri, Joseph Sit, Olga Wodo, Derya Baran, Tayebeh Ameri, Christoph J. Brabec, Adam J. Moule, Baskar Ganapathysubramanian, Nanoscale morphology of doctor bladed versus spin-coated organic photovoltaic films, Adv. Energy Mater. 7 (22) (2017) 1701269.

[27] Raymond T. Hickey, Erin Jedlicka, Balaji Sesha Sarath Pokuri, Adam E. Colbert, Zaira I. Bedolla-Valdez, Baskar Ganapathysubramanian, David S. Ginger, Adam J. Moulé, Morphological consequences of ligand exchange in quantum dot-Polymer solar cells, Org. Electron. 54 (2018) 119–125.

[28] Ryan S. Gebhardt, Pengfei Du, Olga Wodo, Baskar Ganapathysubramanian, A data-driven identification of morphological features influencing the fill factor and efficiency of organic photovoltaic devices, Comput. Mater. Sci. 129 (2017) 220–225.

[29] Pengfei Du, Adrian Zebrowski, Jaroslaw Zola, Baskar Ganapathysubramanian, Olga Wodo, Microstructure design using graphs, Npj Comput. Mater. 4 (1) (2018) 50.

[30] Laurens Van der Maaten, Geoffrey Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (11) (2008).

[31] Koushik Nagasubramanian, Talukder Z. Jubery, Fateme Fotouhi Ardakani, Seyed Vahid Mirnezami, Asheesh K. Singh, Arti Singh, Soumik Sarkar, Baskar Ganapathysubramanian, How useful is active learning for image-based plant phenotyping? 2020, arXiv preprint arXiv:2006.04255.

[32] Martin Arjovsky, Soumith Chintala, Léon Bottou, Wasserstein generative adversarial networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 214–223.

[33] Ameya Joshi, Minsu Cho, Viraj Shah, Balaji Pokuri, Soumik Sarkar, Baskar Ganapathysubramanian, Chinmay Hegde, InvNet: encoding geometric and statistical invariances in deep generative models, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 4377–4384.

[34] Viraj Shah, Ameya Joshi, Balaji Sesha Sarath Pokuri, Sambuddha Ghosal, Soumik Sarkar, Baskar Ganapathysubramanian, Chinmay Hegde, Binary 2D Morphologies of Polymer Phase Separation: Dataset and Python Toolbox, Zenodo, 2019.