Fast inverse design of microstructures via generative invariance networks

Xian Yeow Lee1, Joshua R. Waite1, Chih-Hsuan Yang1, Balaji Sesha Sarath Pokuri1, Ameya Joshi2, Aditya Balu1, Chinmay Hegde2, Baskar Ganapathysubramanian1✉ and Soumik Sarkar1✉

The problem of the efficient design of material microstructures exhibiting desired properties spans a variety of engineering and science applications. The ability to rapidly generate microstructures that exhibit user-specified property distributions can transform the iterative process of traditional microstructure-sensitive design. We reformulate the microstructure design process using a constrained generative adversarial network (GAN) model. This approach explicitly encodes invariance constraints within GANs to generate two-phase morphologies for photovoltaic applications obeying design specifications: specifically, user-defined short-circuit current density and fill factor combinations. Such invariance constraints can be represented by differentiable, deep learning-based surrogates of full physics models mapping microstructures to photovoltaic properties. Furthermore, we propose a multi-fidelity surrogate that reduces expensive label requirements by a factor of five. Our framework enables the incorporation of expensive or non-differentiable constraints for the fast generation of microstructures (in 190 ms) with user-defined properties. Such proposed physics-aware data-driven methods for inverse design problems can be used to considerably accelerate the field of microstructure-sensitive design.

Advances in manufacturing (including additive manufacturing, three-dimensional (3D) printing, layer-by-layer deposition and real-time control) allow us to precisely tailor the spatial distribution (that is, microstructure) of manufactured materials. This opens up the possibility of microstructure-sensitive design, which involves the identification of optimal material configurations that exhibit a desired property. Microstructure-sensitive design can impact a diverse array of applications, ranging from membranes design to enhance water reclamation, battery electrode design to improve energy transport, and organic electronics active-layer design to improve energy harvesting or sensing. However, the systematic creation of fast methods for microstructure-sensitive design is a challenging problem because of the complexity of the ‘forward model’ that maps the microstructure to a property. The availability of a fast ‘inverse design’ framework will transform the field of microstructure-sensitive design, and substantially impact systems that harvest, store and distribute energy and mass.

Over the past decade, a variety of approaches have been explored for solving the inverse microstructure design problem. Traditionally, iterative optimization approaches to search for microstructures that yield desirable characteristics have been in vogue. These approaches are time-consuming, computationally expensive and often require manual supervision by domain experts. Furthermore, such approaches often lack the ability to generalize to new design constraints and require repeated exploration of the design space for each new design constraint. Moreover, optimization-based approaches are susceptible to challenges arising from (1) the combinatorial explosion of plausible microstructures and (2) the computational complexity of function evaluation, that is, solving the forward problem, especially for complex multi-physics problems. Thus, conventional optimization strategies based on using the full physics forward model are a challenging proposition, with limited deployment by groups with the skill set to use them and dedicated access to large computational clusters. As an example, current microstructure optimization approaches that rely on multiple forward-model partial differential equation (PDE) calculations for the discovery of a single optimal (inverse) design typically require up to 160,000 central processing unit (CPU) hours for a given chemical system. The push to democratize microstructure-sensitive design led to efforts focused on relaxing the challenges described above. Some approaches relied on developing cheaper, but less accurate surrogates for the forward model (for example, using graphs), while other approaches made the problem computationally tractable by severely constraining the design space (allowing only specific parameterized shapes).

In this context, recent advances in the field of scientific machine learning show promise for solving inverse design problems. Particularly promising are generative adversarial networks (GANs), a class of generative deep learning model. Given a set of training data points, these models are capable of learning the underlying data distribution to generate new, realistic-looking samples. In the context of engineered systems, GANs have been successfully applied to areas such as differential equations, system modeling and material and drug discovery.

Generative models can be trained to reconstruct realistic-looking microstructures. The challenge is to train them to reconstruct microstructures that satisfy a user-defined set of properties—or, more generally, satisfying a set of constraints. Recently, a modified version of generative model—the invariance network (InvNet)—has been proposed to enable imposition of explicit invariances (or constraints) on the model outputs. The InvNet formulation provides flexibility in terms of incorporating domain knowledge or user design constraints. Specifically, Joshi et al. demonstrated that InvNets can effectively generate two-phase microstructures that satisfy explicit constraints such as volume fractions and domain size and also generate polycrystalline microstructures (a discrete-valued generation problem) by relaxing the problem to a probabilistic assignment problem. However, there are several
drawbacks of the existing InvNet in terms of scalability and when the invariances cannot be explicitly/analytically defined. Additionally, evaluation of invariances is often computationally expensive and time-consuming. For example, evaluating the performance of a given microstructure may involve solving a set of differential equations that can take up to several hours.

This limitation thus hampers the scalability of training InvNet. We propose to represent invariance constraints that cannot be explicitly expressed or are computationally too expensive to evaluate with a deep neural network (DNN) surrogate. By representing the invariance with a surrogate, evaluation of the invariance can be considerably accelerated because the forward evaluation of a neural network is fast once the model is trained. Utilizing a neural network surrogate also has the benefit of not requiring the invariances, such as the equations governing a physical system, to be explicitly known as long as labeled data are available to train the surrogate. Also, a neural network representation of the invariance simplifies the training of InvNet. During training of the InvNet, the parameters of the generative model are optimized by utilizing gradient information from the invariance loss function. Because neural networks are differentiable, gradient information with respect to the invariance loss can be obtained using modern deep learning libraries with automatic-differentiation capabilities. In comparison, using other forms of explicit invariances will necessitate the constraints to be differentiable, and the gradients will have to be calculated separately. Nonetheless, representing the invariance with a neural network does result in a second drawback, which is the availability of labeled data. This motivated us to develop a multi-fidelity surrogate that alleviates the problem of generating expensive labels. In this work, we formulate the microstructure-sensitive design problem into that of training an InvNet with physics-based constraints. We deploy this framework to generate candidate two-phase microstructures for organic photovoltaic (OPV) applications due to the potential of addressing a broad range of problems, as elaborated below.

Flexible, light-weight and wearable electronics and solar cells made from organic components provide a promising solution to a wide array of societal needs. Potential applications of these devices range from sensing (for precision and personalized medicine) to ambient energy harvesting (indoor solar cells) and energy-efficient electronics. For example, the newest generation of small-molecule acceptors have pushed single-junction OPV efficiencies to over 14% and tandem efficiencies to over 17%, potentially revolutionizing green energy harvesting. A large body of work has demonstrated that the morphology in the active layer of OPV devices is key to enabling high-performance devices. Thus, controlling the morphology in the active layer of these devices continues to be crucial for maximizing performance. Advances in the self-assembly of flexible polymers enable remarkable control of hierarchical structure, but the impact on high-performance organic electronics has been limited. This is because, despite the importance of active-layer mesoscale morphology to OPV device performance, it remains a challenge to identify ‘ideal’ microstructures that maximize power conversion efficiencies. A key question is whether multiple ‘families’ of optimal morphology exist, and whether these morphological characteristics depend on material-specific parameters such as electron mobility, exciton diffusion length and bio-molecular recombination. Thus, a rapid, physics-aware microstructure design strategy will enable practitioners to systematically explore and unravel questions of how materials limitations affect optimal morphological features, thereby accelerating materials design leading to high-performance devices.

In this Article, we train an InvNet to generate microstructures that simultaneously obey multiple constraints, specifically user-defined short-circuit current density ($J_{sc}$) and fill factor (FF). These two properties characterize the current–voltage performance of an OPV (for a given material system). The $J_{sc}$ represents the maximum amount of current per unit area that can be drawn across a solar cell (when the applied voltage is zero). The FF denotes the maximum amount of power that can be supplied by the solar cell as a ratio of peak theoretical power. These properties are intimately (and non-trivially) influenced by morphology. We demonstrate that a DNN can be trained as a surrogate model to accurately predict the values of $J_{sc}$ and FF given a two-phase microstructure. This requires a substantial amount of full-fidelity training data, and serves as a baseline forward model surrogate. In the context of our application, creating a labeled dataset of morphologies with corresponding $J_{sc}$ and FF values is computationally expensive, and defeats the goal of avoiding costly physics-based simulations. Hence, we propose a multi-fidelity neural network that achieves a similar predictive accuracy while utilizing a small fraction of full-fidelity labels alongside low-fidelity labels. We then formulate an InvNet-based inverse design framework where these surrogate (high- and multi-fidelity) models are used as invariances to generate morphologies that satisfy user design specifications.

**Results**

We begin with a brief overview of our proposed methodology of using InvNets for the fast generation of targeted two-phase morphologies. Figure 1a illustrates the overall InvNet framework using a Wasserstein GAN (WGAN). The WGAN model formulation ensures that the distribution of generated morphologies matches the true data distribution. Design specifications are enforced via an explicit invariance constraint, whereby the invariance loss is computed using the surrogate physics model represented by a DNN. This invariance loss ensures that the generator produces morphologies that satisfy the invariance constraint. Figure 1b,c displays two alternates to surrogate models: a high-fidelity convolutional neural network (CNN) trained with a large amount of expensive labels obtained from high-fidelity simulations and a multi-fidelity network trained with a mixture of high- and low-fidelity labels. The multi-fidelity network is trained on low-fidelity (but computationally cheap) training labels alongside a fraction of high-fidelity labels so as to reduce the overall computation cost.

We present the following results. First, we validate our CNN surrogate model for accurately predicting the photovoltaic properties of a given morphology, followed by a comparative assessment of the prediction results of the multi-fidelity surrogate model. We then provide illustrative results of the microstructures generated by InvNet using both the high- and multi-fidelity networks.

**High-fidelity short-circuit current and fill factor.** In this section, we validate the approach of utilizing a DNN as a surrogate of the physics-based forward model. This surrogate model also serves as the invariance constraint within the InvNet framework. As DNNs are known to be powerful function approximators with fast prediction times, we hypothesize that they are suitable candidates for representing physics-based forward models. Pokuri et al. previously showed that a CNN can accurately classify microstructures in binned classes of $J_{sc}$. Here, we extend the idea to train a regressor capable of predicting $J_{sc}$ and FF as continuous variables. We train a CNN-based high-fidelity regressor $R_{hi}^s$ on a dataset of two-phase morphology images with high-fidelity simulated $J_{sc}$ values ranging from 0 to 7 mA cm$^{-2}$ and FF values ranging from 0.4 to 0.8 (see Methods and Supplementary section 6 for details on the full physics simulations and material properties). The output of this surrogate model is a vector that consists of the estimated $J_{sc}$ and FF values. Training this model yields $R^2$ values of 0.994 and 0.928 for estimations of $J_{sc}$ and FF, respectively, as seen in the bottom two scatter plots of Fig. 2a. This suggests that the surrogate model is capable of estimating the properties with sufficiently high accuracy. The top two plots in Fig. 2a display histograms of the absolute error for $J_{sc}$ and FF, respectively. The error distribution of $J_{sc}$ has a mean of $0.002 \pm 0.08$ mA cm$^{-2}$ and the error distribution of FF has a mean magnitude of $0.000 \pm 0.02$. 
**Reducing data cost with multi-fidelity labels.** An expected bottleneck of training a surrogate model on high-fidelity labels is the challenge of initial data generation, which may be computationally expensive. Various works have explored the idea of leveraging both high- and low-fidelity data to accelerate computational models. We exploit a similar idea by proposing a multi-fidelity surrogate model to circumvent the challenge of generating computationally expensive high-fidelity labels. First, we provide an overview of computation of the low-fidelity labels, followed by the results of estimating the low-fidelity labels via another neural network surrogate. Then, we present the results of the multi-fidelity surrogate model, which alleviates the requirement for a large, labeled dataset for training. We achieve this by using computationally inexpensive low-fidelity labels and a fraction of expensive high-fidelity labels to train the multi-fidelity surrogate model.

We use our earlier work, where a mechanistic consideration of photophysics as three distinct processes (absorption and generation of excitons; exciton diffusion and dissociation; charge transport and collection) allowed identification of three morphology descriptors that together showed high correlation with $J_{sc}$ (refs. 33,34). These descriptors are computed by representing the two-phase morphology as a weighted graph and evaluating standard graph measures.
(such as connected components and path lengths). Because the complexity of graph-based problems and corresponding algorithms are well understood, these descriptors are computationally inexpensive to compute. These descriptors thus provide a low-fidelity link between morphology and performance. Using only a small fraction of simulated high-fidelity labels ($J_{sc}$ and FF) along with information from low-fidelity labels (morphology descriptors), we train a surrogate model that has similar predictive performance as the model trained purely using high-fidelity labels. Training this multi-fidelity surrogate required the availability of differentiable low-fidelity descriptors, which we get by training another neural network, $R_g$, that maps a morphology to the low-fidelity descriptors (Supplementary sections 2 and 3 present the detailed results).

**Multi-fidelity short-circuit current and fill factor.** We train a multi-fidelity model that estimates $J_{sc}$ and FF using a limited amount of high-fidelity labels with the help of the low-fidelity descriptors. This multi-fidelity network consists of the low-fidelity network, $R_g$, and a separate shared embedding network, as illustrated in Fig. 1c. The purpose of the shared embedding network is to learn additional features that are useful in estimating the properties that were not captured by the low-fidelity model. In our experiments, we used only 20% of randomly sampled high-fidelity labels to train the multi-fidelity network, which resulted in $R^2$ values of 0.989 and 0.894 for $J_{sc}$ and FF, respectively. The absolute error distributions of $J_{sc}$ and FF predictions have means of $-0.009 \pm 0.12$ mA cm$^{-2}$ and $-0.003 \pm 0.02$, respectively. Figure 2b shows the scatter plots of the properties estimated by the multi-fidelity model against the ground-truth values as well as the distribution of errors. As seen in Fig. 2a,b, the $R^2$ values of the $R_{sc}$ and $R_{ff}$ models (HF, high-fidelity; MF, multi-fidelity) are similar, although the label requirements of the multi-fidelity model are reduced by 80%. We stress that, while the low-fidelity network was trained using the entire dataset, the multi-fidelity model was only trained with 20% of the high-fidelity labels, which are more expensive to generate (for example, evaluating the $J_{sc}$ and FF of one morphology needs ~1 CPU h, whereas the low-fidelity metrics can be computed in less than a minute).

Hence, by using the multi-fidelity network, we alleviate the requirement for a large labeled dataset to train a surrogate physics model as the invariance constraint evaluator in the InvNet. For additional results illustrating the performance of the high-fidelity surrogate when trained on a reduced dataset like the multi-fidelity surrogate, see Supplementary sections 3 and 4. It is also worth noting that, for both high- and multi-fidelity models, the prediction accuracy of $J_{sc}$ is noticeably higher than the prediction accuracy of FF. This can be attributed to the fact that FF is a more complex performance feature than $J_{sc}$, as the FF describes the behavior of a photovoltaic device across a voltage range (in contrast to $J_{sc}$, which is evaluated at a fixed voltage). This translates to an expected lower predictive performance of the surrogate model for FF predictions. Nevertheless, we expect that the performance of the surrogate model for FF can still be improved when coupled with optimized network architectures and training hyperparameters.

**Targeted microstructure generation.** We next present the results of generating targeted morphologies that are tailored to design specifications using our proposed InvNet with the multi-fidelity surrogate model framework. Figure 3a show samples of microstructures generated with InvNet for different design specifications. In the top row, we show examples of morphologies with low $J_{sc}$ and high FF values. As we traverse down the rows of Fig. 3a, the specified $J_{sc}$ values increase, while the FF values decrease. We observed that the InvNet-trained generator is able to generate a variety of candidate microstructures with different morphologies given the same design specifications. This indicates that the generator has learned the underlying distribution of the actual data and no mode collapse has occurred during training, which can result in only similar morphologies being generated. This also anecdotally validates a hypothesis in the OPV community that there exist multiple families of morphologies with identical performance. Visualizations of additional samples generated by the InvNet-trained generator for a range of $J_{sc}$ and FF values are provided in Supplementary section 7.

To further verify that the generated morphologies satisfy the imposed design constraints, we generated an additional 1,000

---

**Fig. 2 | Results of the high-fidelity and multi-fidelity surrogate models.** a, Top: error distributions for $J_{sc}$ and FF estimations using the high-fidelity surrogate physics model. Bottom: correlation plots of the estimated properties with respect to the ground-truth values for 9,472 samples with the high-fidelity surrogate physics model. b, Top: error distributions for $J_{sc}$ and FF estimations using the multi-fidelity surrogate model, which was trained with only 20% of high-fidelity labels. Bottom: correlation plots of the estimated $J_{sc}$ and FF values with respect to the ground-truth for the same 9,472 test samples shown in bottom plots of a.
morphologies for different ranges of $J_{sc}$ and FF values and compared the estimated properties of these morphologies with the actual design specifications. The values of these estimated properties and design specifications are plotted as densities and shown in Fig. 3b. We observe that the specified and generated values for both $J_{sc}$ and FF have highly overlapping densities. These overlapping densities confirm that the generator is capable of creating morphologies that satisfy the imposed design specifications, hence enabling targeted design of candidate two-phase microstructures.

Nonetheless, there are situations where the generated morphologies do not adhere to the design specifications, as seen in the first row of Fig. 3b, where the density of generated morphologies (solid green line) has a range of $J_{sc}$ values that are higher than the specified range of $J_{sc}$ values (dashed blue line). Given that this framework is fundamentally data-driven, we hypothesize that this failure mode was caused by an imbalanced dataset where samples from the low $J_{sc}$ and high FF regions might be sparse. To confirm this hypothesis, we visualize the training data distribution in Fig. 3c. Based on the visualization of the joint density, we note that there are indeed very few samples in the top left region, where the morphologies have low $J_{sc}$ and high FF values. However, it is interesting to recognize that, even when the generator fails to generate morphologies with specified $J_{sc}$ in the regions of sparse training data, the rank order of the morphologies' $J_{sc}$ is still preserved. Instead of generating morphologies with random $J_{sc}$ values, the generated morphologies defaulted to morphologies with low $J_{sc}$ and high FF values, which are well supported with data.

**Comparing high-fidelity and multi-fidelity InvNets.** Next, we provide qualitative results to compare the effects of using the high-fidelity ($R_{HF}$) and multi-fidelity ($R_{MF}$) surrogates as the invariance constraint evaluator in InvNet. In Fig. 2, we showed that the performances of the high- and multi-fidelity surrogate models are comparable. Moreover, we are also interested in investigating if the higher variance of the multi-fidelity surrogate will compound and affect the results of the generated morphologies. To study this, we trained InvNet with the same network architecture and replaced the $R_{HF}$ with $R_{MF}$. We illustrate the results from both methods in Fig. 4. In terms of the generated morphologies, we did not observe any notable difference between the two methods. Both the high- and multi-fidelity InvNets are capable of generating microstructures with varying morphologies without signs of mode collapse. However, the density plots that are used to validate the constraint invariances reveal two interesting observations.

First, we observe that the high-fidelity InvNet is more capable of generating low-$J_{sc}$/high-FF morphologies than the multi-fidelity InvNet. This is evident in the first row, where the density of morphologies created by the high-fidelity InvNet is more robust when queried in regions of sparse training data. Comparing high-fidelity and multi-fidelity InvNets. A subsequent high-fidelity InvNet model trained with a larger and more diverse set of morphologies than $R_{HF}$, resulting in the high-fidelity InvNet learning the underlying structure of the low-$J_{sc}$/high-FF morphologies. We attribute this to the fact that $R_{HF}$ was exposed to a much larger and more diverse set of morphologies than $R_{MF}$, resulting in the high-fidelity InvNet learning the underlying structure of the low-$J_{sc}$/high-FF morphologies better when training for the invariance. This suggests that we can expect the performance of high-fidelity InvNet to be more robust when queried in regions where training data are sparser.

The second interesting observation is that the high-fidelity InvNet tends to generate morphologies that are a little more biased in terms of FF. This can be observed in the second, third and fourth rows, where the densities of high-fidelity FF are slightly shifted from the FF design specifications. Referring back to Fig. 3c, we observe that the marginal density of FF data is highly skewed towards the lower regions. Therefore, it is possible that by training $R_{MF}$ on the entire high-fidelity dataset and subsequently using it as the invariance constraint evaluator to train InvNet does result in generated morphologies that are more biased in terms of the design specifications. A subsequent high-fidelity InvNet model trained with a different random initialization also resulted in observations with the same noticeable bias. This highlights the importance of having a balanced dataset when using our proposed framework for morphology generation. This observation can also be interpreted as a
trade-off in the presence of an unbalanced dataset. When the dataset is skewed, training the surrogate model on the entire dataset may yield a generative model that is more robust in sparse-data regions but also generates bias data samples. Conversely, sampling a small fraction of the dataset to train the multi-fidelity surrogate reduces the bias of the surrogate and subsequent generative model, but at the cost of poorer performance when generating samples in regions with less data support.

Efficiency of neural network models versus physics models. In Table 1, we compare the wall-clock running times of our proposed neural network-based methods with physics-based methods for multiple scenarios. All timings were performed on the same platform using a NVIDIA Titan RTX graphics processing unit and averaged across 100 function evaluations. In the first two columns, we show the average computation times for evaluating the $J_{sc}$ and FF properties of a given morphology. We observe that both multi- and high-fidelity methods are several orders of magnitude faster than a high-fidelity physics simulation. A second advantage is that, with the surrogate models, only one evaluation is required to estimate both $J_{sc}$ and FF simultaneously, whereas performing the physics simulation requires separate individual evaluations for $J_{sc}$ and FF.

Comparing $R_{MF}$ with $R_{HF}$, we note that $R_{MF}$ is an order of magnitude faster than $R_{HF}$. However, training $R_{HF}$ comes at the cost of requiring a large dataset with high-fidelity labels. On the other hand, $R_{MF}$ requires a smaller amount of high-fidelity labels, but requires training a more complex model, which increases computation time. Hence, we view the benefits of each method as a trade-off between availability of data and computation time.

In the third column, we show the total time required to train InvNet for $1 \times 10^5$ epochs. We observe that the high-fidelity InvNet is $\sim 3 \times$ faster than multi-fidelity InvNet, which is expected because the training of InvNet is dependent on the surrogate model to compute the invariance loss. We also include an estimate of the time required to train the InvNet if we were to replace the invariance constraint evaluator with an actual physics-based model to compute the invariance loss. As observed, training such an InvNet will require $\sim 60,000$ h, which is not tractable compared to using a neural network-based surrogate model.

Finally, we provide the morphology generation time for a single morphology. Because the process of generating a morphology using InvNet during inference is independent of the surrogate model, there is no notable time difference between using the high- and multi-fidelity InvNet. In summary, we conclude that there is no difference in terms of querying a trained high-fidelity versus multi-fidelity InvNet to generate targeted morphologies. Instead, the deciding factor of which model to use depends on the availability of high-fidelity labels or computation resources. The high-fidelity InvNet framework is faster to train but requires a large dataset of high-fidelity labels to pre-train the surrogate model, whereas the multi-fidelity InvNet model requires fewer high-fidelity labels, but requires a more complex network with longer training times.

Discussion
Although we have demonstrated our proposed framework through the lens of a microstructure design problem that uses a data-driven surrogate, we emphasize that the InvNet framework is certainly not limited to purely data-driven surrogate approaches. Because the invariance constraint of InvNet is explicit, it can be replaced or combined with other data-free approaches. In this regard, a key future direction is to develop InvNets that explicitly incorporate complex physics/domain knowledge in a computationally tractable manner.
These developments are crucial as the proposed framework is currently limited by its dependence on the availability of data and the inability to extrapolate far beyond the support of data. Such limitations are further illustrated in Supplementary section 5. In the context of designing microstructures for OPV devices, one possible approach is to incorporate the photophysics equations governing the microstructure's characteristics as one of the invariance constraints. This may be accomplished by leveraging gradient information that can be queried from the equations. Additionally, results from molecular-scale kinetic modeling that provide best-fit parameterization to experiments can also be incorporated as multi-scale constraints in the InvNet framework.

Furthermore, we briefly expand on the potential implications of our framework for manufacturing and industrial applications. In this work, we have limited our scope to the problem of identifying a set of high-performance morphologies via fast design exploration. That is, we identify the best possible morphologies (conditioned on the performance of the surrogate model) without considering whether such morphologies are currently manufacturable. We argue that such information is useful as it serves to propose target morphologies for manufacturers to achieve by exploring multiple fabrication strategies. However, it is conceptually straightforward to adapt the InvNet framework for designing high-performance manufacturable morphologies. There are multiple possible approaches. A direct approach would be to incorporate additional manufacturability constraints as part of invariance, either via an analytical closed-form expression or through a surrogate model that can be pretrained to characterize the manufacturability of a part.46 Alternately, generative priors47 and additional optimization routines that identify manufacturing conditions that result in the morphology generated by InvNet can also be employed.44 These approaches effectively extend our framework to construct process–structure–property mappings to identify manufacturable designs.

From the OPV perspective, our results show that there are multiple classes of morphologies that exhibit good performance (morphology designs with high $J_{sc}$), as visualized in Supplementary Fig. 9. This is important and useful, as it provides photovoltaic manufacturers more flexibility in terms of morphology design, rather than trying to search for and produce a single hypothesized best morphology. Furthermore, as illustrated by the proposed framework’s capability to explore two different properties, we envision that the same framework can also be deployed by photovoltaic companies to train a model that explores other morphologies on the pareto-front of multiple traits. In this context, another promising future direction includes extending the current framework to generate morphologies with more than two phases as well as validating the generalizability of the framework on a dataset with more than two target properties. To conclude, our vision is that the computational tools developed in this Article will serve to considerably contribute to the area of microstructure-sensitive design for complex multi-functional traits.

### Methods

**Description of two-phase morphology microstructures.** Microstructures. We used a large dataset of microstructure images arising from solving the Cahn–Hilliard (CH) equation with varying initial conditions. The Cahn–Hilliard equation describes phase separation occurring in a binary mixture. It tracks the evolution of the local volume fraction of each phase, in the presence of spatial gradients in the 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 these simulations provide a rich dataset for the design of microstructures.

Specifically, the morphologies obtained through the simulation are visually similar to the bulk heterojunction morphologies observed in organic photovoltaic cells.4 We used an in-house solver for generating the microstructure images. A total of 100 simulations were performed by varying the blend ratio and interaction parameter, $\gamma$ (10 values of $\gamma$ and 10 values of blend ratio). From each simulation, morphologies were obtained every 20 time steps to ensure distinguishable morphologies across time steps. Each simulation took around 2 hours using eight CPUs on a high-performance computing (HPC) cluster. More details about the computations and computational efficiency of the numerical method are provided in refs. 12–14. These simulations led to a diverse set of morphologies representative of the bulk heterojunction morphologies typically seen in solution-processed OPV devices. After curating and ensuring that each morphology was unique, we obtained a collection of 34,672 microstructure images, each with a resolution of 101 x 101 pixels (resized further to 128 x 128 pixels for use in CNNs). Each image is grayscale, with the value of each pixel ranging between 0 and 1.

**Photophysics annotation of microstructures.** Each morphology was virtually interrogated to extract its current–voltage characteristics by solving a morphology-aware (that is, spatially heterogeneous) photophysics device model. We deployed a validated, in-house software that uses a finite element-based solution strategy for solving the 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 PDEs that model the optoelectronic physics of energy harvesting in OPV devices.

The photophysics consists of the following stages:

- Incident solar radiation causes the generation of energetically active electron–hole pairs, called excitons (denoted by $X$), in the donor regions of the microstructure. These excitons diffuse across the microstructure and have a finite lifetime before becoming ground-state electron–hole pairs.

- Excitons that diffuse and reach the donor–acceptor interface undergo dissociation into electrons (denoted by $n$) and holes (denoted by $p$) at the donor–acceptor interface. The dissociation mechanism is material- and field-dependent (denoted by $D$).

- These generated charges diffuse through the microstructure and reach their corresponding electrodes (cathode and anode) to produce a current. Two mechanisms are responsible for driving carrier transport or current flow: (1) drift, which is caused by the presence of an electric field (denoted as the gradient of the potential, $\nabla \phi$) and (2) diffusion, which is caused by a spatial gradient of electron or hole concentration.

- The distribution of electrons and holes in the microstructure interacts with the applied voltage and influences the electrostatic potential $\phi$ across the microstructure. Finally, electrons and holes can recombine (denoted by $R$) to create excitons.

This photophysics is encoded as the XDD equations:

\[
\nabla J_n - R + D = 0 
\]

\[
-\nabla J_p + R + D = 0 
\]

\[
\nabla (\epsilon_{\text{d}} \nabla \phi) = q(n - p) 
\]

\[
-\nabla \left( V_{\text{bi}} \mu_n \nabla X \right) - \beta_D \nabla^2 \phi - R_{\text{bi}} = -G - R_{\text{bi}} 
\]

where $X$, $n$, and $p$ represent the exciton, electron, and hole distributions, respectively, $\phi$ represents the electric potential, $q$ represents the elementary charge, $V$ represents the thermal voltage, $\epsilon$ is the dielectric constant in the donor and recipient materials, and $\mu_{\text{d/h}}$ are electron/hole/exciton mobilities, respectively. The current densities $J_n$ and $J_p$ are given by the constitutive equations

\[
J_n = -q \mu_n \frac{\nabla \phi}{\epsilon} + q V_{\text{bi}} \mu_n \nabla n 
\]

\[
J_p = -q \mu_p \frac{\nabla \phi}{\epsilon} - q V_{\text{bi}} \mu_p \nabla p 
\]
These sets of high-dimensional, complex PDEs are solved to obtain the performance of the solar-cell device, which is characterized by the short-circuit current and fill factor.

Training details. High-fidelity surrogate model. To improve the robustness of the surrogate model, we performed standard image augmentation techniques, image rotation and flipping, which resulted in a dataset of ~38,000 images of augmented morphologies. To ensure a stable training process, we scaled the labels of \( L \) and FF to belong in the same numerical range. Following standard practices, we partitioned 80% of the data as training data and reserved 20% of data as test data. Because the task of the surrogate model is to essentially perform a multi-target regression, the loss function of the regressor is formulated as

\[
L_{\text{reg}} = \| R_{\text{FF}} (I) - L \|^2 + \| R_{\text{FF}} (I) - FF \|^2
\]

where \( R_{\text{FF}} \) denotes the high-fidelity surrogate model, parameterized by parameter \( \phi \), is the input image of the microstructure and \( J \) and FF are the true label values. The output of \( R_{\text{FF}} \) is a vector of two values that correspond to the estimated \( J \) and FF values. The model was trained using the Adam optimizer with a learning rate of 3 \times 10^{-5} for 25 epochs.

Multi-fidelity surrogate model. Before describing the training details, we briefly justify the need to replace the graph-based computation of low-fidelity descriptors with another neural network surrogate, \( R_s \), in the multi-fidelity model. Although multi-fidelity frameworks are effective at reducing the requirement for expensive labels, they are not inherently tractable for application as an invariance constraint in InvNets. This is because updating the generator's parameters in InvNet requires the gradient computation of the invariance loss function. However, the graph-based methods used to compute the low-fidelity descriptors are often non-differentiable. Therefore, optimizing the parameters of the generator via conventional back-propagation becomes a non-trivial problem. Additionally, evaluating the low-fidelity descriptors using the previously proposed graph-based methods requires that the generated images be converted into nodes and edges on the fly during training, which incurs additional computational cost and time. Accordingly, a neural network surrogate that is differentiable and can directly evaluate the graph features of morphologies in the pixel domain circumvents both challenges.

As illustrated in Fig. 1c, the multi-fidelity network encompasses both a low-fidelity network and a shared embedding network. The purpose of the shared embedding network is to learn additional features that are not already captured by the low-fidelity network for estimating \( J \) and FF. During training of the multi-fidelity network, the low-fidelity network predicts the low-fidelity descriptors of a given microstructure, which are combined with the embeddings from the shared embedding network. These two vectors are then passed through a dense layer to estimate \( J \) and FF. As we are only using a limited amount of high-fidelity labels, it is possible that training the multi-fidelity network might lead to a biased model due to label imbalance. To avoid such issues, we constructed the following weighted loss function with empirically determined scaling constants that balances the errors between the estimations of \( J \) and FF. \( R_s \) denotes the multi-fidelity surrogate model, where \( \gamma \) and \( \omega \) represent the parameters of the shared embedding network and low-fidelity network, respectively:

\[
L_{\text{reg}} = L_s + \omega \cdot L_F
\]

\[
L_s = \lambda_1 (L_s^2 + \omega) + \left( |R_{\text{MF}} (I) - J|^2 \right) \]

\[
L_F = \lambda_2 |R_{\text{MF}} (I) - FF|^2
\]

with \( \lambda_1 \) and \( \lambda_2 \) heuristically set to 0.008 and 0.0005, respectively. We highlight that, in principle, the weights of the low-fidelity network \( R_s \) are already trained and can be frozen. Nevertheless, in practice, we find that allowing the weights of the low-fidelity network to be optimized alongside the entire network does result in slightly better estimations. To train the multi-fidelity network, we used the stochastic gradient descent (SGD) optimizer with a learning rate of 1 \times 10^{-4} and trained the network for 100 epochs.

Generator and discriminator of InvNet. In this section, we provide the training description of InvNet with the multi-fidelity surrogate model as the invariance constraint evaluator. Because the main modification that we proposed occurs in the invariance constraint, the formulation of InvNet’s loss function remains as

\[
L_{\text{InvNet}} = L_G (\theta, \psi) + L_D (\theta)
\]

where \( L_G \) denotes the standard loss function of the WGAN, \( \theta \) are the parameters of the generator and \( \psi \) are the parameters of the discriminator. Both the generator and discriminator are also represented using DNNs. The invariance loss \( L_I \) is expressed as

\[
L_I = \| R_{\text{inv}} (G_s(z)) - R_{\text{inv}} (G_l(z)) \|^2 + \| R_{\text{inv}} (G_s(z)) - R_{\text{inv}} (G_l(z)) \|^2
\]

with \( G_s \) denoting the generator, \( z \) a latent vector sampled from a uniform distribution, \( G_l(z) \) the generated morphology and \( I \) a real morphology sampled from the dataset. During training, the weights of the surrogate model, \( \varphi \), are kept frozen, and \( R_{\text{inv}} \) acts purely as an invariance constraint evaluator that estimates the properties of the generated microstructures. Only the parameters of the discriminator, \( \psi \) and generator, \( \theta \), are optimized.

To train the InvNet, we instantiate the generator with an architecture that consists of one dense layer, five residual blocks with skipped connections, and one convolution layer. Each residual block is made up of two batch-normalization layers and two convolution layers with up-sampling operations. Rectified linear unit (ReLU) activation functions were used after every layer, except for the last convolution layer. We used the sigmoid activation function on the output of the convolution layer to generate 128 × 128 images of microstructures. The discriminator network consists of one convolution layer, four residual blocks and a fully connected layer. The residual blocks are similar to the blocks used in the generator, with the exception that the convolution layers are paired with down-sampling operations and layer-normalization is used instead of batch-normalization. As we have chosen to use the WGAN-Gradient Penalty (WGAN-GP) variant of GAN, the output of the discriminator is a single scalar value estimating the Wasserstein distance between the distributions of the real and generated microstructures. This variant of GAN helps robustly estimate the generator mode against possible mode collapse failures during training. To compute the invariance loss, we use the multi-fidelity surrogate model \( R_{\text{inv}} \) to ensure that generated morphologies has properties that are similar to the properties of the real morphologies. Both the generator and discriminator are trained alternately using the Adam optimizer with a learning rate of 1 \times 10^{-4} for 10^5 epochs. The discriminator and generator are trained for more epochs with a lower learning rate than for the surrogate models because the generator's model training process is a form of zero-sum game and involves the generator and discriminator losses reaching an equilibrium. Thus, a lower learning rate helps stabilize the training process. For the training of all the models presented in this study, the number of training epochs is decided based on the same principle where we stop training once a satisfactory performance was observed. Note that, in the methodology presented, we have described the InvNet framework using the multi-fidelity surrogate, \( R_{\text{inv}} \), as the invariance constraint evaluator. We highlight that the methodology for training the framework using a high-fidelity network is the same, with only \( R_s \) replacing \( R_{\text{inv}} \).

Model architectures. InvNet architecture. In Supplementary Table 2, we provide the details of the parameters used in the generator and discriminator models, respectively. The residual blocks in the generator consist of two sequential pairs of batch-normalization and convolutional layers. Therefore, in each residual block is coupled with an up-sampling operation while the second convolution layer is a normal convolution layer. Note that, in every residual block, there is also a skip connection that is parameterized by a convolution layer. The output of each residual block is computed by performing an element-wise addition between the output convolution features and skip-connection features. The output of the final convolution layer in the generator is a grayscale 128 × 128 image of a morphology. In the discriminator, the residual blocks consist of two sequential pairs of layer-normalization and convolutional layers. The first convolution layer in each residual block is paired with a down-sampling operation while the second convolution layer is a normal convolution layer. There is also a skip connection in every residual block that is parameterized by a single convolution layer. The output of the final dense layer in the discriminator is a single scalar value representing the Wasserstein distance.

High-fidelity regressor architecture. Next, we provide details on the network architecture used to train the high-fidelity surrogate model, \( R_{\text{HF}} \). The model consists of the following layers in sequential order: two convolution blocks (each having a convolution, batch-normalization, ReLU activation and max pooling layers) and two dense layers. Dropout layers with a dropout rate of 0.3 were also used between every convolution-block and dense layers during training to avoid over-fitting. The output of the final dense layer comprises the estimated value of \( J \) and FF, respectively. The parameters of each layer are provided in Supplementary Table 3.

Multi-fidelity regressor architecture. The multi-fidelity regressor, \( R_{\text{MF}} \), consists of two sub-networks, the shared embedding network and the low-fidelity network, as illustrated in Fig. 1c. As shown in Supplementary Fig. 2a, the low-fidelity network consists of two branches. The top branch consists of three convolutional and max pooling layers, while the bottom branch consists of two dense layers. The outputs of these two branches are concatenated together into a single branch, which contains three additional dense layers. ReLU activation functions are used between each layer. The outputs of the low-fidelity network are the estimates of the low-fidelity descriptors of a given morphology. The parameters for each layer are tabulated in Supplementary Table 4. Empirically, we observed that the convolutional branch is helpful in learning features to estimate \( g_1 \), and the dense branch is helpful in estimating the values of \( g_2 \) and \( g_3 \).

We now focus on the architecture of the shared embedding network, which consists of two interconnected branches as depicted in Supplementary Fig. 2b.
The first branch (bounded by the top dotted-line box, labeled branch 1) consists of an initial convolution layer, $Conv_{1s}$ (colored gray), and a dropout layer with a rate of 0.1. The branch then splits into two identical sub-branches (shown in blue and orange) that each consist of a convolution layer, max pooling, dropout and two more convolution layers. The output of the first sub-branch is combined with the predictions from the low-fidelity network and passed through a final dense layer to predict the values of $f_{r}$. The second branch (bounded by the second dotted-line box at the bottom of Supplementary Fig. 2b, labeled branch 2) consists of three convolution layers each followed by a pooling layer and a dropout layer, as well as an additional convolution layer. The outputs of the second main branch are combined with the predictions of the low-fidelity network along with the outputs from the second sub-branch of branch 1. These concatenated outputs are passed through two more dense layers to estimate the values of the FF. Supplementary Table 4 tabulates the parameters used for each layer in the shared embedding network.

Data availability

The data generation procedure for the dataset used in this study is explained in the Methods of the manuscript, and the generated dataset is available from the repository in ref. 

Code availability

The codes for generating the dataset are available in the following code repository: . The data and code used to train the models are available in a code capsule and the code used to generate the figures and results of this study is available in a separate code capsule . Source data are provided with this paper.

Received: 6 October 2020; Accepted: 22 February 2021; Published online: 25 March 2021

References

1. Anthony, J. E. Addressing challenges. Nat. Mater. 13, 773–775 (2014).
2. Yao, H. et al. A flexible and highly pressure-sensitive graphene–polyurethane sponge based on fractured microstructure design. Adv. Mater. 25, 6692–6698 (2013).
3. Du, P., Zebrowski, A., Zola, J., Ganapathysubramanian, B. & Wodo, O. Microstructure design using graphs. NPJ Comput. Mater. 4, 50 (2018).
4. Noruzi, R., Ghadai, S., Krishnamurthy, A. & Ganapathysubramanian, B. NURBS-based microstructure design for organic photovoltaics. Comput. Des. 118, 102771 (2021).
5. Pokuri, B. S. S., Gholash, S., Kokate, A., Sarkar, S. & Ganapathysubramanian, B. Interpretable deep learning for guided microstructure–property explorations in photovoltaics. NPJ Comput. Mater. 5, 99 (2019).
6. Stoecklein, D., Lore, K. G., Davies, M., Sarkar, S. & Ganapathysubramanian, B. Deep learning for flow sculpting: insights into efficient learning using scientific simulation data. Sci. Rep. 7, 46368 (2017).
7. Lore, K. G., Stoecklein, D., Davies, M., Ganapathysubramanian, B. & Sarkar, S. A deep learning framework for causal shape transformation. Neural Netw. 98, 305–317 (2018).
8. Bock, F. E. et al. A review of the application of machine learning and data mining approaches in continuum mechanics materials. Front. Mater. 6, 110 (2019).
9. Lee, Y. Y., Balu, A., Stoecklein, D., Ganapathysubramanian, B. & Sarkar, S. A case study of deep reinforcement learning for engineering design: application to microfluidic devices for flow sculpting. J. Mech. Des. 141, 111401 (2019).
10. Balu, A. et al. A deep learning framework for design and analysis of surgical bioprosthetic heart valves. Sci. Rep. 9, 18560 (2019).
11. Geng, Y., van Anders, G., Dodd, P. M., Dishmehchase, J. & Glotzer, S. C. Engineering entropy for the inverse design of colloidal crystals from hard shapes. Sci. Adv. 5, eaaw0514 (2019).
12. Kim, B., Lee, S. & Kim, J. Inverse design of porous materials using artificial neural networks. Sci. Adv. 6, eaax9324 (2020).
13. Goodfellow, I. et al. Generative adversarial nets. In Advances in Neural Information Processing Systems 2672–2680 (NIPS, 2014).
14. Yang, L., Zhang, D. & Karniadakis, G. E. Physics-informed generative adversarial networks for stochastic differential equations. SIAM J. Sci. Comput. 42, A292–A317 (2020).
15. de Oliveira, L., Paganini, M. & Nachman, B. Learning particle physics by example: location-aware generative adversarial networks for physics synthesis. Comput. Softw. Biomed. 21, 1–4 (2017).
16. Wu, J.-L. et al. Enforcing statistical constraints in generative adversarial networks for modeling chaotic dynamical systems. J. Comput. Phys. 406, 109209 (2020).
17. Yang, Z. et al. Microstructural materials design via deep adversarial learning methodology. J. Mech. Des. 140, 114146 (2018).
18. Sanchez-Lengeling, B. & Aspuru-Guzik, A. Inverse molecular design using machine learning: generative models for matter engineering. Science 361, 360–365 (2018).
Articles

45. Balaji. pbbsarath/excitonic_drift_diffusion: v1.0, https://doi.org/10.5281/zenodo.4552777 (2021).

46. Lee, X. Y. et al. Fast inverse design of microstructures via generative invariance networks (training) (2021); https://www.codeocean.com/, https://doi.org/10.24433/CO.1250930.v1

47. Lee, X. Y. et al. Fast inverse design of microstructures via generative invariance networks (evaluation and figures) (2021); https://www.codeocean.com/, https://doi.org/10.24433/CO.3688453.v1

Acknowledgements
This work was supported by the ARPA-E DIFFERENTIATE programme under grant no. DE-AR0001215. B.G., C.-H.Y. and B.S.S.P. were supported in part by DoD MURI 6119-ISU-ONR-2453. C.H. and A.J. were supported in part by NSF grants CCF-2005804 and CCF-1815101. We thank A. Krishnamurthy and Z. Bao for fruitful discussions and constructive suggestions. Computing support from XSEDE and Iowa State University is also gratefully acknowledged.

Author contributions
C.H., B.G. and S.S. initiated the project; B.S.S.P. and B.G. planned and generated the dataset. X.Y.L., A.J., C.H., B.G. and S.S. designed the machine learning framework. X.Y.L., J.R.W. and C.-H.Y. performed the training. X.Y.L., A.J. and A.B. analyzed the data. All authors contributed to writing the manuscript.

Competing interests
The authors declare no competing interests.

Additional information
Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s43588-021-00045-8.

Correspondence and requests for materials should be addressed to B.G. or S.S.

Peer review information Nature Computational Science thanks Samuel Cooper and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Jie Pan was the primary editor on this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

© The Author(s), under exclusive licence to Springer Nature America, Inc. 2021