SUPPLEMENTARY MATERIAL FOR FAST INVERSE DESIGN OF MICROSTRUCTURES VIA GENERATIVE INVARIANCE NETWORKS

1 Architectures

In the following sub-sections below, we provide model architecture details for the neural networks that we used in the main manuscript.

1.1 InvNet architecture

In Tables S1, we provide the specific details of the parameters used for each layer in the Generator and Discriminator respectively. As described in the main manuscript, the Residual blocks in the Generator consists of two sequential pairs of batch-normalization and convolutional layers with the parameters tabulated in top section of Table S1. The first convolution layer 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 blocks, there is also a skip-connection that is parameterized by a single convolution layer. The output of each Residual block is finally 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 consists of two sequential pairs of layer-normalization and convolutional layers with parameters shown in bottom section of Table S1. 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.

[Table 1 about here.]

1.2 High-fidelity regressor architecture

Next, we provide details on the network architecture that was used to train the high-fidelity surrogate model, R_{HF}. The model consists of the following layers in sequential order: two convolution blocks (each having a convolution layer, batch normalization layer, ReLU activation, and a max pooling layer) and two dense layers. Dropout layers with a dropout rate of 0.3 were also used in between every convolution block and dense layers during training to avoid over-fitting. The output of the final dense layer is the estimated values of J_{sc} and FF respectively. The parameters of each layers are provided in Table S2.

[Table 2 about here.]

1.3 Multi-fidelity regressor architecture

The mult-fidelity regressor, R_{MF} consist of two sub-networks, the shared embedding network and the low-fidelity network, as illustrated in Figure.1(c) of the main text.

We begin with the description of the low-fidelity network's architecture. As seen in Figure S1(a), the low-fidelity network consist of two branches. The top branch consists of three convolutional and max-pooling layers, while the bottom branch consists of two dense layers. The output of these two branches are concatenated together into a single branch, which contains three additional dense layers. ReLU activation functions are also used in between every layer. The output of the low-fidelity network are the estimates of the low-fidelity descriptors of a given morphology. The parameters for each layers are tabulated in Table S3 for reference. Empirically, we observed that the convolutional branch is helpful in learning features to estimate $g1$ while the dense branch is helpful in estimating the values of $g2$ and $g3$

We now focus on the architecture of the shared embedding network, which consists of two interconnected branches as depicted in Figure S1(b). The first branch (bounded by the top dotted-line box, labeled Branch 1) consists of an initial convolution layer, $Conv_H$(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 yellow) that both 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 J_{sc}. The second branch (bounded by the second dotted line box at the bottom of Figure S1(b), labeled Branch 2) consists of three convolution layers, each followed with max-pooling and a dropout layer, and a 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 another two dense layers to estimate the values of the FF. Table S3 tabulates the parameters used for each layer in the shared embeddings network.

[Figure 1 about here.]

[Table 3 about here.]

2 Additional results on multi-fidelity experiments

2.1 Details on training low-fidelity model

As described in the main text, training the multi-fidelity surrogate model requires the availability of differentiable low-fidelity descriptors. Following in line with the paradigm of leveraging the capacity of neural networks as function approximators, we trained another neural network, R_g, that maps a morphology to the low-fidelity descriptors. As the computation of the low-fidelity descriptors are computationally cheaper than the high-fidelity labels, we trained R_g on the entire dataset of 38k pairs of images and descriptors.

The low-fidelity surrogate R_g network architecture consists of two blocks that runs in parallel, a convolutional block with convolutional layers and a dense block with only fully-connected layers. The output of these two blocks are then concatenated through another dense block to predict a vector of the three low-fidelity descriptors as illustrated in Figure S1(a). Since this results in a regression formulation, the loss function is expressed as:

$$L_{R_g} = \|R_{\omega,g1}(I) - g1\|_2^2 + \|R_{\omega,g2}(I) - g2\|_2^2 + \|R_{\omega,g3}(I) - g3\|_2^2 \tag{1}$$

where ω denotes the parameters of the deep neural network and $g1, g2$ and $g3$ denotes the ground truth values of the low-fidelity descriptors. We trained the low-fidelity surrogate model with Stochastic Gradient Descent (SGD) optimizer with a learning rate of 1E-3 for 250 epochs. During training, the magnitudes of $g1$, $g2$ and $g3$ were scaled according to their minimum and maximum values so that all three descriptors fall within the same numerical range.

2.2 Results from low-fidelity model's training

The results of training R_g to estimate the low-fidelity descriptors of a morphology are provided in Figure S2. We observe that the estimates and true values of $g1$ form an almost perfect fit with along the diagonal, signifying that the low-fidelity model is capable of estimating the first morphological descriptor accurately. On the other hand, the estimates of the $g2$ and $g3$ predicted by the model are more varied, as compared to estimates of $g1$. Nevertheless, all three estimations had high R^2 values with $g1$, $g2$ and $g3$ having an R^2 of 0.998, 0.977 and 0.976 respectively. While the performance of this low-fidelity model is sufficient in training the overall multi-fidelity model, we hypothesize that the architecture and hyper-parameters of this low-fidelity can definitely be further optimized to decreased the variances of the estimated $g2$ and $g3$.

[Figure 2 about here.]

2.3 Effects of different ratios of data

We provide additional results for an ablation study conducted using the multi-fidelity model. After pre-training the low-fidelity model to estimate the low-fidelity descriptors accurately, we performed an ablation study on the effect of high-fidelity labels by progressively decreasing the amount high-fidelity labels in the training set from 70% of to 1%. As the primary aim of the multi-fidelity model within the InvNet framework is to reduce the amount of high-fidelity labels required, it is essential to understand the limits of the multi-fidelity model's performance as a function of high-fidelity training data size. This ensures that we are able to appropriately reduced the amount of high-fidelity labels without disrupting the training process of InvNet significantly. Table S4 summarizes the results of the ablation study. We observed that the R^2 values for both J_{sc} and FF decreases as we decrease the amount of high-fidelity labels, which is expected of a data-driven model. More importantly, we note that the R^2 values experiences a sharper decrease when the labels were decreased from 20% and below. Hence, we determined that the 20% threshold is an appropriate amount of high-fidelity labels for training a multi-fidelity surrogate model that has similar performance to the high-fidelity model while reducing the required number of expensive high-fidelity labels.

[Table 4 about here.]

3 Out-of-distribution experiments

In this section, we show the behavior of the trained InvNet outside the support of training distribution. Since the proposed InvNet is entirely data-driven, it is imperative to understand how InvNet will perform when queried for morphologies that are outside the training data distribution, e.g., generate morphologies with both high J_{sc} and FF values. Since such morphologies do not exist in the training dataset (both in the training set of the surrogate physics model as well as the training set of InvNet), generating such morphologies will require InvNet to extrapolate to previously unseen regions of the data distribution.

Figure S3 shows four examples of such scenarios. In Figure S3(a), we queried for microstructures with low J_{sc} and FF values. Since microstructures with such properties do exist in the training data, albeit sparsely, InvNet

was able to generate the specified microstructures. However, observe that the property densities of the generated microstructure have slightly shifted peaks and are not as uniformly distributed as the specified densities. Figure S3(b) illustrates a more extreme case where we queried InvNet for microstructures with high values of J_{sc} and FF. Such microstructures do not exist in the dataset. As observed, InvNet fails to generate such microstructures and instead generated microstructures with properties in ranges that are supported by data. Additionally, Figures S3(c) and (d) represent slightly relaxed extrapolation scenarios, where we queried InvNet for microstructures with one property being well supported by training data while the other property being out-of-distribution. In Figure S3(c), we queried for microstructures with J_{sc} that are out of the training distribution but kept the design specifications of FF within the range of training data. That resulted in generated microstructures with FF values that were close to the specified FF, while the J_{sc} values of the generated microstructures defaulted to the values ranging around 8, which is the range of highest J_{sc} values present in the training data. Last but not least, Figure S3 illustrates InvNet's behavior when queried for microstructures with moderate J_{sc} but high FF values. Here, we observed that the generated microstructures had J_{sc} values relatively closer to the design specifications of J_{sc}, as compared to the FF property, which was far from the specified FF values.

Collectively, we have observed that the reliability of the performance does decrease when queried for samples that are out of the training data distributions since InvNet was trained entirely based on data. This is a limitation of our proposed method and does highlight the emphasis required for hybrid methods, which learns from data and domain-based knowledge in future works.

[Figure 3 about here.]

4 Full physics interrogation of designed microstructures

[Figure 4 about here.]

We evaluate the performance of a set of high performing morphologies (i.e. high J_{sc}) produced by the InvNet. Three morphologies from this set are shown in Figure. S4(b). We compute the photovoltaic performance of these morphologies by solving the full physics XDD equations and plot the current voltage curve in Figure. S4(a). The photovoltaic properties evaluated from the full physics simulation is very closely matched by that predicted by the InvNet as iilustrated in Figure. S4(c). This suggests that design exploration can be rapidly performed using the InvNet to identify a very small subset of promising morphologies. These morphologies can then be evaluated using the full physics simulator, dramatically reducing design time and cost.

5 Visualization of morphologies in low dimensional embedding space

To obtain a clearer insight into the distribution of morphologies, we perform a dimensional reduction on the images of microstructures and visualize the low-dimensional embeddings. We achieve this by flattening every 128 x 128 image of microstructure in the training dataset into a 16384 feature vector. We then project these feature vectors onto a two-dimensional plane using Uniform Manifold Approximation and Projection (UMAP) [1]. The results of the projection is shown in Figure S5(a), using a hyper-parameter of 25 neighbours to perform the embeddings. The embeddings of each morphology image is color-coded according to the magnitude of their short-circuit current J_{sc}. From Figure S5(a), we observed that morphologies with high magnitudes of J_{sc} do form distinctive clusters while morphologies with lower magnitudes ($\approx$ 0-2 mA/m^2) of J_{sc} form a single large central cluster. Furthermore, visualization of the morphologies from the distinctive clusters with high-performing J_{sc} reveals that each cluster has a distinctive form of geometry. This anecdotally confirms the hypothesis that there exist several families of high-performing morphologies.

In Figures S5(b) and (c), we project samples of microstructures with a wide range of J_{sc} values generated by the high-fidelity and multi-fidelity InvNet using the same projection mapping learnt for embedding the training data. The projection of these generated data points(colored) are superimposed onto the embeddings of the training data(gray) for a clearer comparison. In both visualizations, we observe that the embeddings of the generated microstructures do follow the distributions of training data embeddings. Another interesting observation is that both versions of InvNets also generated microstructures with high J_{sc} values in the embedding regions where the microstructures in the training dataset only have medium J_{sc} values. We hypothesize that this is attributed to InvNet's generalization capabilities which resulted in the generation of novel morphologies which have similar geometry to morphologies with medium-ranged J_{sc} yet actually have high magnitudes of J_{sc}.

[Figure 5 about here.]

References

[1] Leland McInnes, John Healy, and James Melville. "Umap: Uniform manifold approximation and projection for dimension reduction". In: *arXiv preprint arXiv:1802.03426* (2018).

List of Figures

S1 **Illustrations of network architectures for multi-fidelity surrogate model. (a)** Architecture of low-fidelity Network that is initially pre-trained before being used in the multi-fidelity network. **(b)** Illustration of the multi-fidelity network, consisting of the shared embedding network (shown in grey, blue and yellow) and the low-fidelity network (green). 5

S2 **Results from training the low-fidelity model R_{LF} to estimate the low-fidelity descriptors of a morphology.** Observe that the model is capable of estimating all three descriptors with high accuracy, although the variance of $g2$ and $g3$ estimates are a little higher than $g1$. 6

S3 **Exploration of Invnet's behaviour in regions that are out of the training data's support for (a) low J_{sc} and FF (b) high J_{sc} and FF (c) high J_{sc} and mid FF and (d) mid J_{sc} and high FF.** Scenarios **(a)** and **(b)** represents more extreme extrapolations where the data is sparse for both J_{sc} and FF while **(c)** and **(d)**represents a slightly relaxed extrapolation where data is available for one property but sparse for the other, i.e. mid-range J_{sc} which is available in the training data but high-range FF which is not present in the training data as shown in **(d)**. 7

S4 **Results of full physics simulation for InvNet-designed microstructures. (a)** JV-Curves of the top three high performing morphologies (high J_{sc}) sampled from the set of morphologies generated by InvNet. **(b)** Morphologies corresponding to JV curves shown in **(a)**. **(c)** Comparison of J_{sc} values obtained from full physics simulation of 35 Invnet-generated morphologies with the J_{sc} predicted values by the surrogate model. 8

S5 **Visualization of morphologies in low dimensional embedding space**. **(a)**Visualization of morphologies in the entire training dataset projected onto a two-dimensional plane via UMAP with 25 nearest neighbours. We observe that morphologies with high magnitudes of J_{sc} do form distinctive clusters. Additional visualizations reveal that each clusters contains microstructures with a specific type of geometry. **(b)** UMAP visualization of microstructure samples generated by high-fidelity InvNet and **(c)** microstructure samples generated by multi-fidelity InvNet super-imposed on training data. Observe that the generated microstructures closely follows the distribution of the training data in the embedding space. Microstructures with high J_{sc} values were also generated in embedding regions where the training data only contains microstructures with medium-ranged J_{sc}. 9

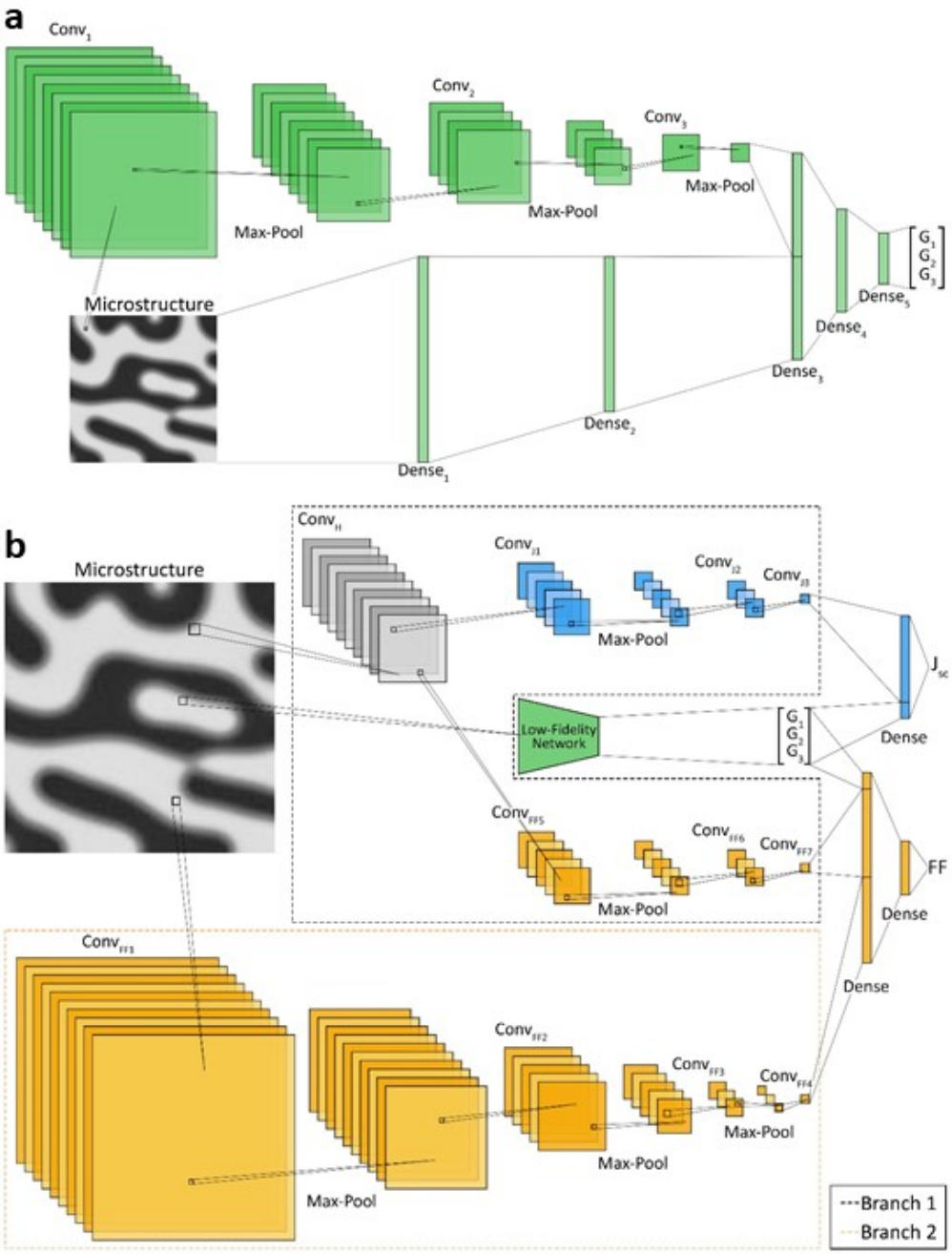

Figure S1: **Illustrations of network architectures for multi-fidelity surrogate model. (a)** Architecture of low-fidelity Network that is initially pre-trained before being used in the multi-fidelity network. **(b)** Illustration of the multi-fidelity network, consisting of the shared embedding network (shown in grey, blue and yellow) and the low-fidelity network (green).

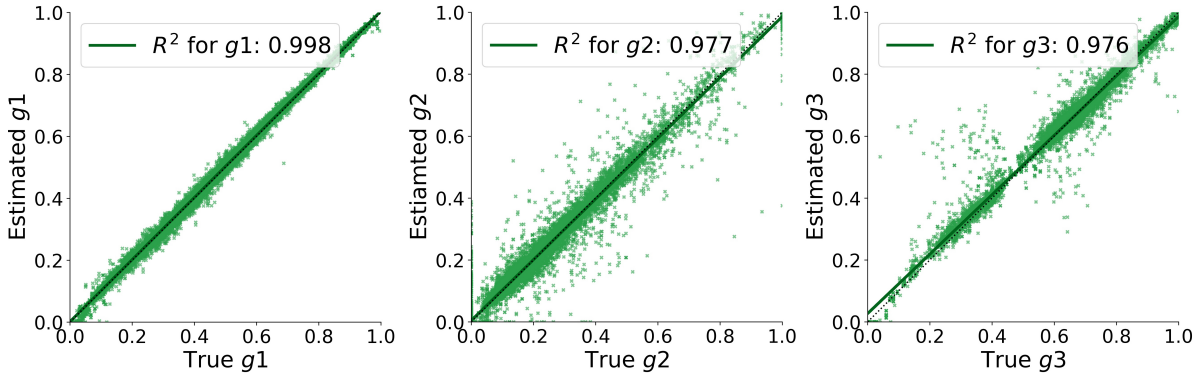

Figure S2: **Results from training the low-fidelity model R_{LF} to estimate the low-fidelity descriptors of a morphology.** Observe that the model is capable of estimating all three descriptors with high accuracy, although the variance of $g2$ and $g3$ estimates are a little higher than $g1$.

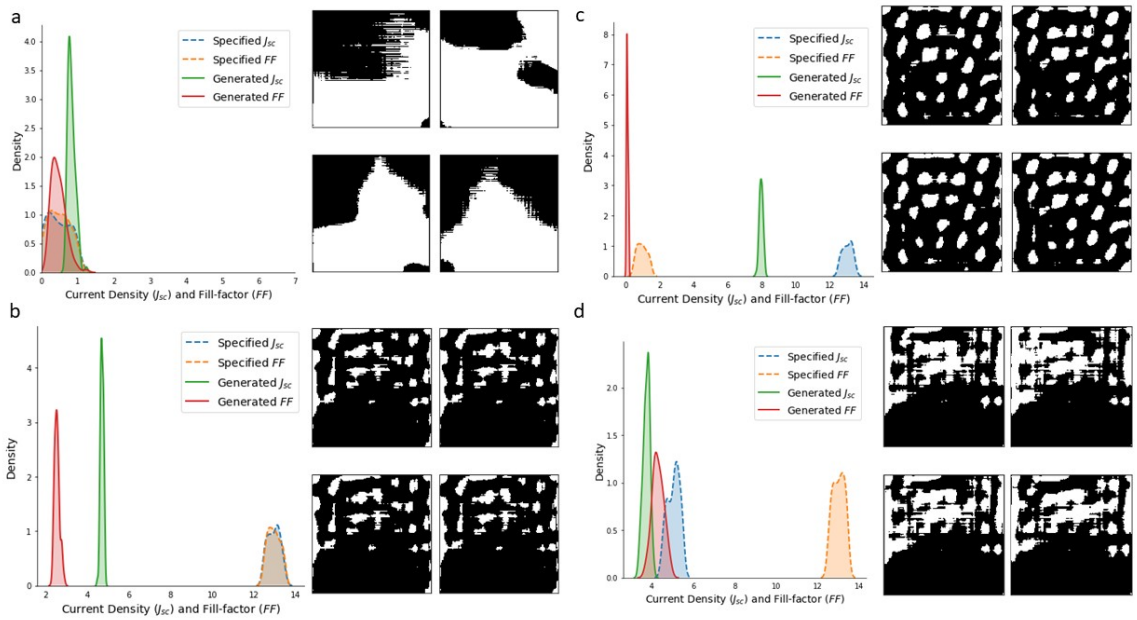

Figure S3: **Exploration of Invnet's behaviour in regions that are out of the training data's support for (a) low J_{sc} and FF (b) high J_{sc} and FF (c) high J_{sc} and mid FF and (d) mid J_{sc} and high FF.** Scenarios **(a)** and **(b)** represents more extreme extrapolations where the data is sparse for both J_{sc} and FF while **(c)** and **(d)**represents a slightly relaxed extrapolation where data is available for one property but sparse for the other, i.e. mid-range J_{sc} which is available in the training data but high-range FF which is not present in the training data as shown in **(d)**.

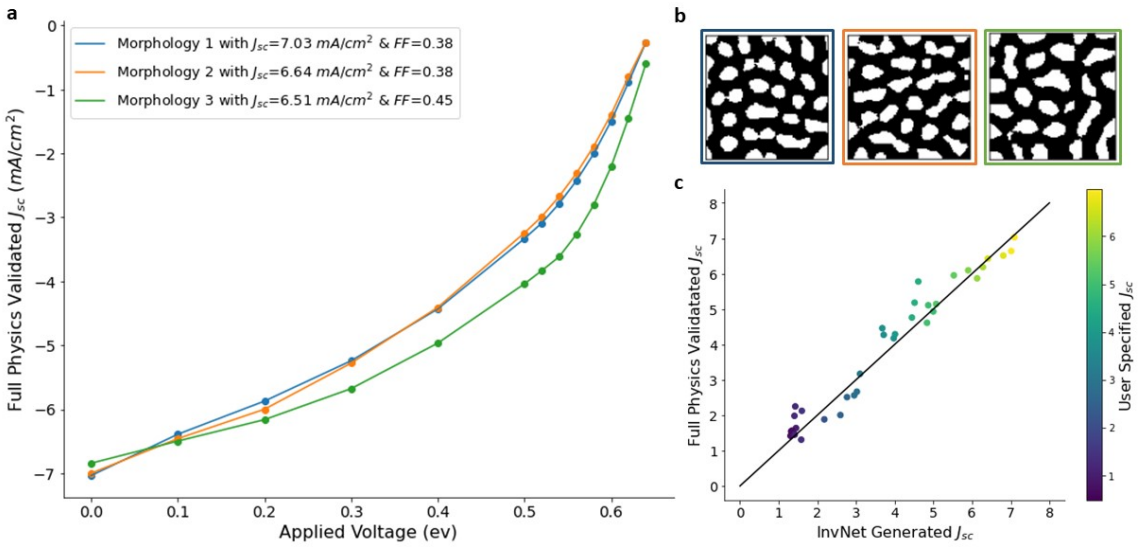

Figure S4: **Results of full physics simulation for InvNet-designed microstructures. (a)** JV-Curves of the top three high performing morphologies (high J_{sc}) sampled from the set of morphologies generated by InvNet. **(b)** Morphologies corresponding to JV curves shown in **(a)**. **(c)** Comparison of J_{sc} values obtained from full physics simulation of 35 Invnet-generated morphologies with the J_{sc} predicted values by the surrogate model.

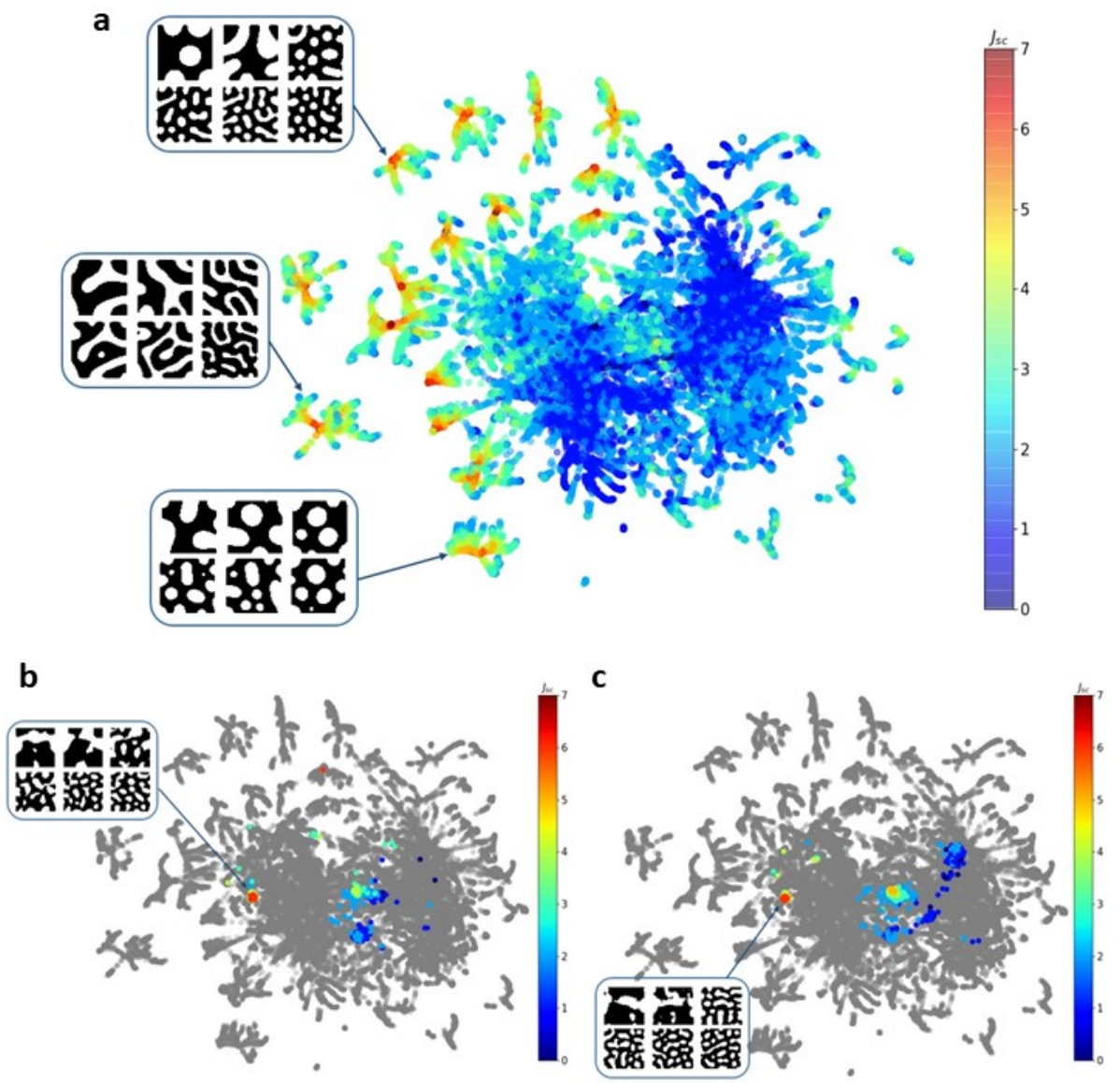

Figure S5: **Visualization of morphologies in low dimensional embedding space**. **(a)**Visualization of morphologies in the entire training dataset projected onto a two-dimensional plane via UMAP with 25 nearest neighbours. We observe that morphologies with high magnitudes of J_{sc} do form distinctive clusters. Additional visualizations reveal that each clusters contains microstructures with a specific type of geometry. **(b)** UMAP visualization of microstructure samples generated by high-fidelity InvNet and **(c)** microstructure samples generated by multi-fidelity InvNet super-imposed on training data. Observe that the generated microstructures closely follows the distribution of the training data in the embedding space. Microstructures with high J_{sc} values were also generated in embedding regions where the training data only contains microstructures with medium-ranged J_{sc}.

List of Tables

S1 **Number of input features, output features, kernel size and stride for each layer used in the Generator and Discriminator's network.** 11

S2 **Number of input features, output features, kernel size and stride for each layer used in high-fidelity surrogate model R_{HF}.** 12

S3 **Parameters of layers used in the low-fidelity network and in the shared embeddings network.** 13

S4 **Ablation study results performed using multi-fidelity network to study the effects of reducing high-fidelity training data labels.** Observe that the decreasing trends for both R^2 values are mostly linear until the 20% threshold, upon which the R^2 starts to decrease more sharply 14

	Generator			
	Input Features	Output Features	Kernel Size	Stride
Dense Layer	128	16384	-	-
Residual Block 1	1024	1024	3	1
Residual Block 2	1024	512	3	1
Residual Block 3	512	256	3	1
Residual Block 4	256	128	3	1
Residual Block 5	128	128	3	1
Convolution Layer	128	1	3	1
	Discriminator			
	Input Features	Output Features	Kernel Size	Stride
Convolution Layer	1	128	3	1
Residual Block 1	128	256	3	1
Residual Block 2	256	512	3	1
Residual Block 3	512	1024	3	1
Residual Block 4	1024	1024	3	1
Dense Layer	65536	1	-	-

Table S1: **Number of input features, output features, kernel size and stride for each layer used in the Generator and Discriminator's network.**

Layer Name	Input Features	Output Features	Kernel Size	Stride
Conv_1	1	16	9	1
Conv_2	16	32	9	1
Dense_1	21632	8192	-	-
Dense_2	8192	2	-	-

Table S2: **Number of input features, output features, kernel size and stride for each layer used in high-fidelity surrogate model R_{HF}.**

	Low Fidelity Network			
Layer Name	Input Features	Output Features	Kernel Size	Stride
Conv_1	1	128	3	3
Conv_2	128	64	3	1
Conv_2	64	1	3	1
Dense_1	16384	2048	-	-
Dense_2	2048	1024	-	-
Dense_1	1024	256	-	-
Dense_2	512	256	-	-
Dense_1	256	64	-	-
Dense_2	64	3	-	-
	Shared Embedding Network			
Layer Name	Input Features	Output Features	Kernel Size	Stride
Conv_H	1	20	4	4
Conv_J1	20	10	2	2
Conv_J2	10	5	3	1
Conv_J3	5	1	2	2
Dense_J	19	1	-	-
Conv_FF1	1	20	3	1
Conv_FF2	20	10	2	2
Conv_FF3	10	5	3	2
Conv_FF4	5	1	3	1
Conv_FF5	20	10	2	2
Conv_FF6	10	5	3	1
Conv_FF7	5	1	2	2
Dense_FF1	35	10	-	-
Dense_FF2	10	1	-	-

Table S3: **Parameters of layers used in the low-fidelity network and in the shared embeddings network.**

% of Training Data	R^2 for J_{sc}	R^2 for FF
70%	0.994	0.947
50%	0.992	0.935
40%	0.992	0.928
30%	0.990	0.916
20%	0.989	0.894
10%	0.981	0.854
5%	0.971	0.817
1%	0.944	0.682

Table S4: **Ablation study results performed using multi-fidelity network to study the effects of reducing high-fidelity training data labels.** Observe that the decreasing trends for both R^2 values are mostly linear until the 20% threshold, upon which the R^2 starts to decrease more sharply