Multi-objective Evolutionary Design of Microstructures using Diffusion Autoencoders

Anirudh Suresh[1] **Devesh Shah**[2] **Alemayehu Admasu**[2]
Devesh Upadhyay[2] **Kalyanmoy Deb**[1]
[1]Michigan State University [2] Ford Motor Company
{suresha2,kdeb}@msu.edu

Abstract

Efficient design of microstructures with targeted properties has always been a challenging task owing to the expensive and time-consuming nature of the problem. In recent years, generative models have been used to accelerate this process. However, most of these methods are hindered by the choice of their generative model - either due to stability and usability, like with GANs, or flexibility of the model itself, like the availability of a semantically meaningful latent space. We propose a diffusion autoencoder based generative design framework that not only provides the fidelity and stability benefits of diffusion models but also has a desirable latent space that can be exploited by evolutionary algorithms. We employ this framework to solve multiple simultaneous objectives to find a Pareto frontier of candidate microstructures. We also show that the search space of optimization can be drastically reduced by conditioning the model with target objective values. We demonstrate the efficacy of the proposed framework on a number of optimization and generative tasks based on two-phase morphology dataset derived from Cahn-Hilliard equations.

1 Introduction

The task of designing materials that possess certain target properties, called *inverse problem* or the *material discovery problem*, is a challenging and tedious procedure. The aim here is usually to identify optimal process parameters that can lead to materials with target properties. Recent advances in manufacturing technologies, such as additive manufacturing, have paved the way for a new type of inverse problem, called microstructure-sensitive design problem, where the aim is to design the microstructure of the material rather than the process parameters. While this gives more control with respect to the design process, this also leads to more challenges due to the size of the search space and the complexity of forward models. Constraining the search space [1] or use of surrogate models [2] can alleviate some of these issues but microstructure design problems still remain challenging.

Recently, the use of deep generative models (DGMs) for solving inverse problems have shown remarkable success. Use of these DGMs, owing to their ability to generate realistic data, have been thoroughly explored in literature for solving of inverse problems [3]. This process, called generative design (GD), focuses on training DGMs with available data and driving generation toward desired outcomes using optimization tools. Most existing methods in the material discovery domain [4, 5, 6, 7, 8, 9] use generative adversarial networks (GANs) as their choice of DGMs. These methods are severely hindered by their choice of DGMs due to well-known deficiencies of GANs. GANs are notoriously difficult to train and are known for their instability and model collapse issues. Diffusion models (DM) [10] are another family of DGMs that have outperformed GANs in generative capabilities. Diffusion models are considered stable alternatives to GANs as they retain the fidelity benefits while being easy to train and use. Some existing generative design methods have adapted DMs as the choice of their DGMs [11, 12]. Herron et al. [13] and Dureth et al. [14] showed superior

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

capabilities of denoising diffusion probabilistic models (DDPM) on microstructure datasets. However, previously discussed generative design methods cannot be readily retrofitted with diffusion models and the use of DMs for generative design in material discovery is relatively unexplored and needs to be studied.

For a GD task to be tractable, we need the DGM to have superior generative capabilities and be easy to train and use. Owing to the nature of datasets in domains like material discovery, it can be difficult to select optimally trained DGMs as typical metrics such as FID and Inception score might not be meaningful [15]. Due to the finite amount of data and lack of visually interpretable features, it can also be difficult to detect failures such as mode collapse or unwanted artifacts. We also need a semantically meaningful latent space that can be used during optimization or interpolation. A meaningful latent space ensures that nearby solutions are *similar* to one another leading to a promising optimization landscape. For example, if a near-optimal solution found by our framework but turns out to be infeasible to manufacture, this candidate solution cannot be discarded as a similarly optimal yet feasible solution might be available nearby. The ability to search locally and find closely related solutions is a much-needed characteristic for successful GD. While vanilla DDPM/DDIM models are the clear choice for their fidelity and stability benefits, they do not have a latent space that can be used for optimization. Hence, we propose a diffusion autoencoder model [16] based optimization framework that can be readily exploited by evolutionary algorithms (EA) for single- and multi-objective optimization [17]. We use genetic algorithms (GA) for sigle-objective optimization and NSGA-II [18] for multi-objective optimization. Since diffusion autoencoders are conditioned on both semantic and stochastic subcodes, given a candidate semantic subcode provided by the optimization algorithm, multiple stochastic subcodes can be sampled to create *similar* microstructures, that can circumvent issues with artifacts or manufacturability of the obtained solutions. Another important avenue for a GD framework is constrained sampling. The proposed optimization framework can be readily modified to accommodate conditioning on target objectives to drastically reduce the search space and speed up the optimization process. While most existing studies have only focused on the quality of generated candidates, we approach this problem from an optimization perspective and rigorously evaluate the proposed framework on single and multi-objective optimization problems with varying difficulty based on two-phase morphology dataset [4] derived from Cahn-Hilliard equations. By framing a variety of modified datasets and targets, we emulate various practical scenarios commonly encountered while solving material discovery problems.

The reminder of the study is arranged as follows - in Section 2, we discuss existing literature. In Section 3, we describe our proposed GD framework and in Section 4, we describe the dataset and optimization problems used in this study. In Section 5, we discuss results and finally in Section 6, we summarize our findings and propose future directions of research.

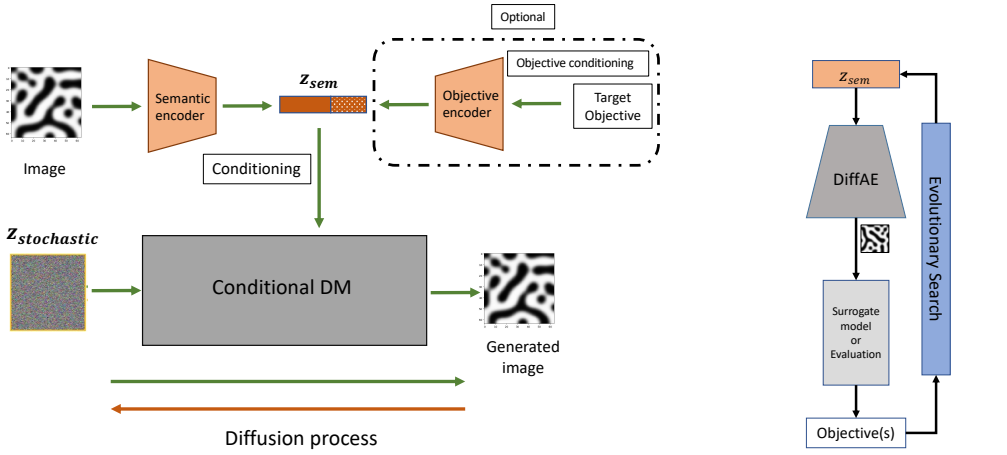
2 Related works

GD methods have gained a lot of attention in the recent past. These methods have been used to solve a variety of inverse problems, such as topology optimization [11, 12, 19, 20], materials discovery material science [5, 6, 8, 4, 9], design [21, 22] etc. We refer readers to [3] for a detailed review of the use of DGMs for optimization. GANs have been extensively used for microstructure design as microstructures are typically represented as 2D images and conveniently fit into the strengths of GANs. Yang et al. [5] train GANs and use Bayesian optimization (using noise vectors) for solving the inverse problem. Fokina et al. [6] use StyleGAN [23] architecture for generating realistic microstructure designs. Chen and Liu [7] use a conditional GAN to generate microstructures by using a geometry-aware loss function. Tan et al. [8] perform inverse optimization using a DCGAN coupled with a convolutional neural network (CNN) based surrogate model. Lee et al. [4] propose an inverse optimization framework based on encoding invariance constraints on the GANs. Similarly, Lambard et al. [9] used StyleGAN2 [24] coupled with ADA for generating high-resolution microstructures. Most of these methods are restricted by the choice of their DGM and cannot be readily retrofitted with stable alternatives like DMs due to the nature of the generation procedure. A few recent works have adapted DM architectures for microstructure generation - Herron et al. [13] use DDPMs for designing two-phase morphology microstructures and demonstrate the superior performance of diffusion models compared to WGANs [25]. Similarly, Dureth et al. [14] illustrate the generative capabilities of DMs by using class conditioning. The use of diffusion models along with a focus on GD in the materials discovery domain is scarce and needs immediate attention.

3 Proposed generative design framework

3.1 Diffusion autoencoder architecture

Diffusion models have shown remarkable success in generative tasks. These models learn an iterative denoising process to decode an arbitrary standard Gaussian noise prior to a target clean image. Song et al. [26] proposed a non-markovian variant of DDPMs called denoising diffusion implicit models (DDIM) that uses a deterministic generative process thereby improving sampling efficiency. The latent space of DDIMs is not suitable for performing optimization or meaningful interpolation as they consist of Gaussian noise. Preechakul et al. [16] proposed a conditional DDIM architecture that consists of an additional semantic encoder, $z_{sem} = \text{Enc}_\phi(x_0)$ that learns to map the input image x_0 to a semantically meaningful latent subcode z_{sem}. Hence, the DDIM decoder is conditioned on both a semantic latent subcode, z_{sem}, and a stochastic subcode that consists of Gaussian noise, $z_{stochastic}$ (x_T). Here, z_{sem} is a non-spatial vector of $d = 128$ similar to a style-vector in StyleGAN [23, 24] architecture. The diffusion autoencoder (DiffAE) architecture is shown in Figure 1a.



(a) Conditional and unconditional DiffAE

(b) Optimization loop

Figure 1: Proposed framework

3.2 DiffAE based optimization framework

For successful GD and design, or solving inverse problems, the following properties are desirable:

1. **Generative capabilities:** The DGM needs to be able to create realistic and feasible candidate solutions that are outside the realm of training data. The DGM needs to be able to innovate in order to find promising candidates.
2. **Ease of use:** The DGM needs to be easy to train and use as the dataset might be sparse, ill-conditioned, or small.
3. **Semantic search space:** The latent space used for optimization should be meaningful and exhibit *consistency* properties. That is, two solutions close to each other in latent space should be similar in features or objectives or both.
4. **Conditioning:** The DGM should also have scope to condense the search space by using additional conditioning.

The choice of DGM architecture is not trivial as the lack of these properties can severely hinder GD capabilities. GANs, as demonstrated by literature, have the needed generative capabilities along with desirable search space and conditioning ability. However, they are not easy to train and use. Similarly, vanilla DDPM/DDIM architectures have shown great success in fidelity, ease of use, and conditioning capabilities but do not have a meaningful latent space. DiffAE architecture on the other hand provides all the aforementioned benefits making this the optimal choice for GD and inverse problems.

In our proposed GD framework, shown in Figure 1b, z_{sem} of diffusion autoencoders will act as decision variables for an evolutionary algorithm (EA). For every candidate z_{sem}, x_T is randomly sampled. Evolutionary methods are a convenient choice here as they are known to be effective for global optimization. However, other methods like Bayesian optimization can also be incorporated here instead of the EAs.

3.3 Target conditioning

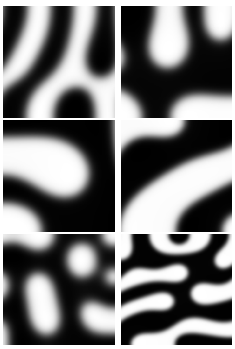
We extend the DiffAE architecture to have additional conditioning from an objective encoder (MLP in this study) that encodes target objective values, as shown in Figure 1a. This way, the relationship between the objective values and the generated images is learned and can be later exploited using conditional sampling. This method is similar to class conditional sampling but we condition using continuous values. Now, instead of performing global optimization from scratch, target conditioning can help reduce the search space and expedite the optimization procedure.

It can be tempting to replace the optimization procedure with conditional sampling, as conditional sampling can give us the desired candidates. Instead of *searching* for optimal candidates, objectives can be provided for conditioning and the generated candidate can be used as the optimal candidate. However, we argue that conditioning might not be as effective in all parts of the objective space and lack of information regarding uncertainty at different regions of the objective space will make this method unreliable. Hence, we propose to use target conditioning merely to reduce the search space, that is, to help start optimization at a *near optimal* solution and perform further optimization as a fine-tuning step. This removes the uncertainty regarding the *accuracy* of target conditioning.

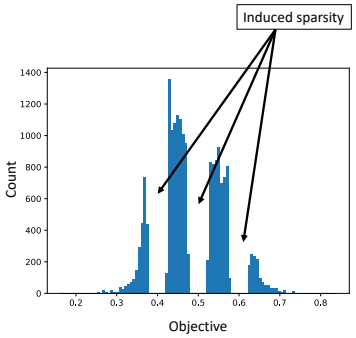
4 Problem definition

4.1 Two-phase morphology dataset

We use two-phase morphology dataset [27] derived from solving Cahn-Hilliard (CH) equation [28] with varying initial conditions. The CH equations represent the phase separation in a binary mixture and track the volume fraction of the phases. These images derived from CH simulations hold visual similarities to bulk heterogeneous morphologies observed in organic photovoltaic cells [29]. These synthetic microstructures have been widely used as surrogates for microstructure-sensitive design problems [4]. The dataset used here consists of a total of 38,578 grayscale images, each with a resolution of 128 x 128 pixels and values in the range of 0 and 1. A few examples are shown in Figure 2a.



(a) Training data



(b) Modified datasets

Figure 2: Two-phase morphology dataset

Given microstructures, objectives such as fill factor (FF) and short circuit current (J_{sc}) can be computed by solving steady excitonic drift diffusion equations (XDD) equations. We refer readers to [4] for more details on computing these objectives. We use the following objectives for framing GD problems:

- **One point correlation (P_1):** Computed as the mean of the pixel values and represents the volume fraction of the binary mixture.
- **Fill factor (FF):** The maximum amount of power that can be supplied by the solar cell as a ratio of peak theoretical power.
- **Short circuit current (J_{sc}):** The maximum amount of current that can be drawn across the solar cell per unit area.

Here, P_1 is directly influenced by the pixel values and features while FF and J_{sc} are non-trivially influenced by the morphology [30]. These objectives act as an effective test bench for evaluating GD frameworks.

4.2 Optimization problems

While most studies in this domain have focused on the quality of the newly created samples, we frame a number of generative and optimization tasks to understand the strengths of the proposed framework. In order to emulate different practical aspects of GD, we modify the datasets by inducing sparsity and inefficiencies. The modified datasets are of the following categories:

- ***Sparse:*** The datasets are made incrementally sparse in order to test the stability, generalizability, and ease of use aspect of the DGMs. We induce sparsity in the objective space by removing chunks of data as shown in Figure 2b. We frame four levels of sparsity that span the whole domain with respect to features and objective values but have gaps and discontinuities. They represent a common practical scenario of sparse datasets collected using experiments spanning the entire search space.
- ***Half:*** We consider the extreme case of removing half of all available data with respect to the objective space. These datasets are *incomplete* with respect to features and objective values. We use these datasets to evaluate the optimization capability of the proposed framework by setting up optimization problems where the optimum is incrementally moved away from the available data. These problems are of increasing difficulty in nature as the target to be achieved is further away from the available data and represents a need to innovate in order to find optimal microstructures. The target optima are shown in Table 1 where *limit* represents the maximum objective value present in the modified training dataset. This is a direct emulation of an inverse problem where the DGM needs to design microstructures that embody the target objective values.

The amount of data present in each modified dataset is reported in supplementary material.

Table 1: Problems with increasing difficulty levels for DMs trained on half datasets

	Limit	Target 1	Target 2	Target 3	Target 4
P_1	0.5	0.55	0.6	0.65	0.7
FF	0.1415	0.2	0.25	0.3	0.35
J_{sc}	0.3314	0.4	0.45	0.5	0.55

Apart from these single-objective problems, we create a two-objective problem where both FF and J_{sc} need to be maximized. This is based on the computation of power conversion efficiency (PCE) [31] as shown in Equation 1. We use all available data for training DGMs for multi-objective optimization due to the inherent difficulty of multi-objective optimization.

$$PCE = \frac{FF.J_{sc}.V_{oc}}{P_{in}/A} \tag{1}$$

where, V_{oc} is the open-circuit voltage, P_{in} is the input power of incident light and A is the active area of the photo-voltaic module.

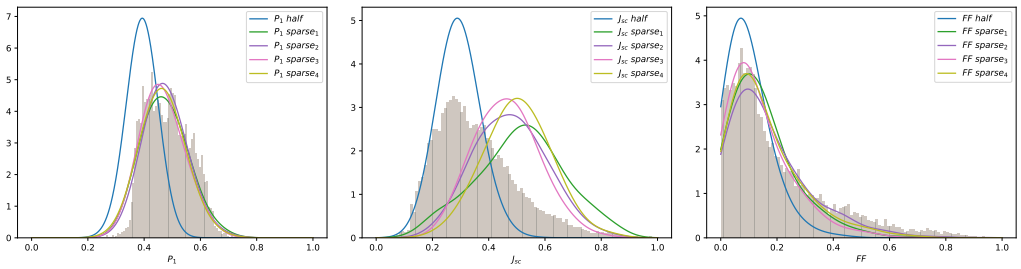
5 Results

ResNet18 based surrogate models were used for evaluating FF and J_{sc}. These surrogate models achieved R^2 values of 0.997 and 0.97 on the test set respectively. More details regarding architectures

and training hyperparameters are provided in the supplementary document. Evolutionary algorithms are implemented using pymoo[1] [32] and DiffAE models are based on the original implementation [2].

5.1 Generative capability

First, we evaluate the generative capability of the DGMs by training them on modified datasets and analyzing the distribution of objective values of generated samples. We randomly sample 1000 z_{sem} vectors from the latent space of the models (based on bounds of latent vectors of training data) and generate microstructures and evaluate their objective values. The aim is to understand the effect of the induced sparsity on DGM training from the perspective of objectives. Figure 3 shows density plots of objective values evaluated from unconditional generation of different DiffAE models. The Histogram of the original dataset is shown as a shaded region in each plot. One observation that is common to all three objectives is that the *half* dataset models yield substantially different densities compared to the other sparse datasets. As expected, these span a little more than the limit of the data (from Table 1) but drop rapidly as no data is available in those regions. We investigate this further using results from single-objective optimization problems. On the other hand, we can observe that the sparsity of the data does not affect the distribution of objectives learned by the DGMs. Irrespective of the sparsity of the data, the learned distributions span the whole domain. This shows that the DGM has filled the gaps in the dataset via interpolation of features as needed. This gives us confidence that this architecture is suitable for GD as sparse data is often encountered during practical applications. This helps us derive valuable insight regarding collecting data for GD - sparse yet complete datasets are better to work with than dense and incomplete datasets. Gaps and inconsistencies in features and objectives can be overcome by the generative model.



(a) P_1 (b) J_{sc} (c) FF

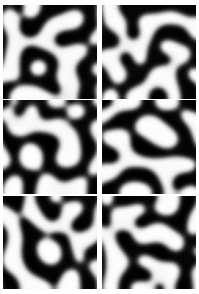
Figure 3: Unconditional generation

Generating multiple versions of the same candidate microstructure can be beneficial from the perspective of feasibility, removing unwanted artifacts or manufacturability. We can achieve this in our proposed framework by keeping the semantic subcode constant (z_{sem}) and varying only the stochastic subcode (x_T). This will generate multiple renderings of the same image, with similar features and minor differences. An example of this is shown in Figure 4 where Figure 4a shows the generated microstructures and Figure 4b shows their corresponding objective values (P_1). These microstructures are *similar* and have comparable P_1 values.

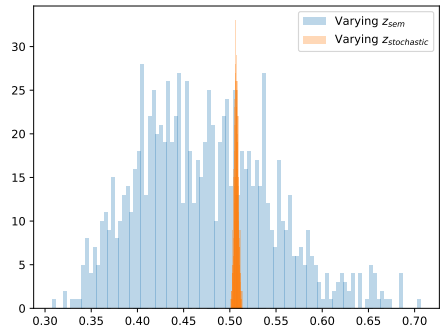
5.2 Single objective optimization

Single objective GAs are run for 200 generations with a population size of 50, leading to a total of 10k function evaluations (surrogate evaluations for J_{sc} and FF). All optimization experiments are performed 31 times and median results are presented. Values not present in the dataset are set as targets and L_1 error from the target is used as the objective for minimization. The median performance of the GAs at the first and last generation of optimization are presented in Table 2 and the corresponding progress is presented in Figure 5 with solid lines. The first objective considered is P_1 which is directly influenced by the pixel values. The first case is the $sparse_1$ problem where the target is set in the middle of the induced sparsity. We can see that the GA finds corresponding latent vectors that can generate microstructures with the correct volume fractions. The increasing difficulty

[1] https://github.com/anyoptimization/pymoo
[2] https://github.com/phizaz/diffae



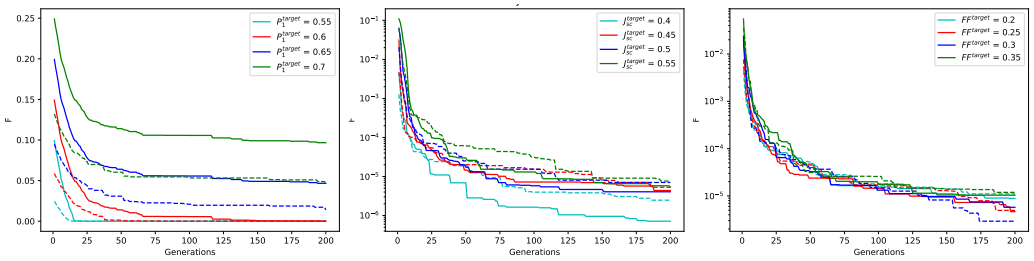
(a) Variation in generated microstructures



(b) Comparison of P_1 while varying z_{sem} versus varying $z_{stochastic}$ and keeping z_{sem} constant

Figure 4: Effect of keeping the semantic subcode (z_{sem}) constant while only changing the stochastic subcode ($z_{stochastic}$) for P_1 $sparse_1$ DGM

of the other problems is demonstrated by the achieved objective values. As the target is moved further away from the training dataset, the generated samples are less optimal, as seen in Figure 5a. This problem is also visually intuitive - as the target amount of white pixels increases, the DGM is unable to extrapolate and create candidates with corresponding P_1 values.



(a) P_1 (b) J_{sc} (c) FF

Figure 5: Median performance of EAs - solid lines represent unconditional optimization and dashed lines represent target objective conditioned optimization

On the other hand, optimization performance for FF and J_{sc} are drastically different from the P_1 case. These objectives are non-trivially influenced by morphology and hence, further away in the objective space might not imply further away in feature space. This is demonstrated in Figure 5b and Figure 5c where all problems with different difficulty levels perform similarly and achieve similar final error values. Even though the microstructures corresponding to target objective values are not present in the training dataset, the GAs find latent vectors that can generate optimal microstructures.

5.3 Multi-objective optimization

For multi-objective optimization, we need to maximize both FF and J_{sc}. NSGA-II is employed for this with a population size of 50 and 600 generations, giving a total of 30k solution evaluations. This experiment is run 31 times and median results are presented.

The aim here is to find a Pareto frontier of candidate microstructures that are *better* than the microstructures in the available dataset. The final PO front achieved by the framework for the median run is shown in Figure 6. A few microstructures along the Pareto front are also shown. Here, the blue points indicate the final PO front and the gray points indicate the training dataset. We can observe that most of the generated candidates dominate the training dataset. Now, these non-dominated solutions can be further analyzed to choose a desirable solution by the decision maker.

Table 2: Median performance of GAs at first and last generation

Dataset	Objective	Target	Start (Gen = 1)		End (Gen = 200)	
			Unconditional	Conditional	Unconditional	Conditional
$sparse_1$		0.5	2.27E-03	**9.70E-05**	2.98E-07	2.38E-07
$half$		0.55	9.91E-02	**2.46E-02**	2.98E-07	5.36E-07
$half$	P_1	0.6	1.49E-01	**5.88E-02**	1.62E-04	3.24E-05
$half$		0.65	1.99E-01	**9.49E-02**	4.67E-02	1.39E-02
$half$		0.7	2.49E-01	**1.32E-01**	9.67E-02	4.60E-02
$sparse_1$		0.3314	6.90E-03	5.89E-03	2.95E-06	4.86E-06
$half$		0.4	4.37E-03	**1.28E-03**	7.15E-07	2.47E-06
$half$	J_{sc}	0.45	3.15E-02	**4.77E-03**	4.38E-06	5.30E-06
$half$		0.5	6.05E-02	**2.05E-02**	4.05E-06	7.03E-06
$half$		0.55	1.09E-01	**6.55E-02**	5.36E-06	7.63E-06
$sparse_1$		0.1415	2.72E-03	2.20E-03	5.28E-06	6.90E-06
$half$		0.2	5.26E-03	4.42E-03	8.85E-06	1.11E-05
$half$	FF	0.25	1.97E-02	**7.65E-03**	4.65E-06	4.95E-06
$half$		0.3	2.38E-02	2.28E-02	5.75E-06	2.92E-06
$half$		0.35	5.52E-02	3.60E-02	1.04E-05	1.19E-05

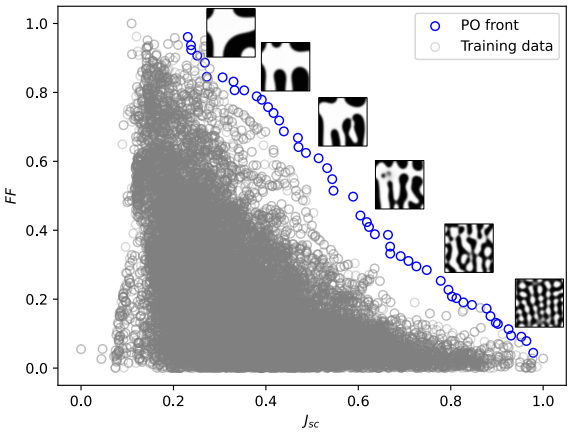
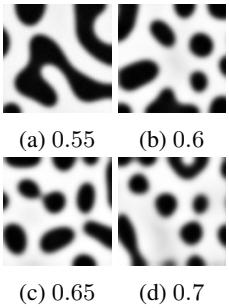


Figure 6: PO front along with the training data



(a) 0.55 (b) 0.6

(c) 0.65 (d) 0.7

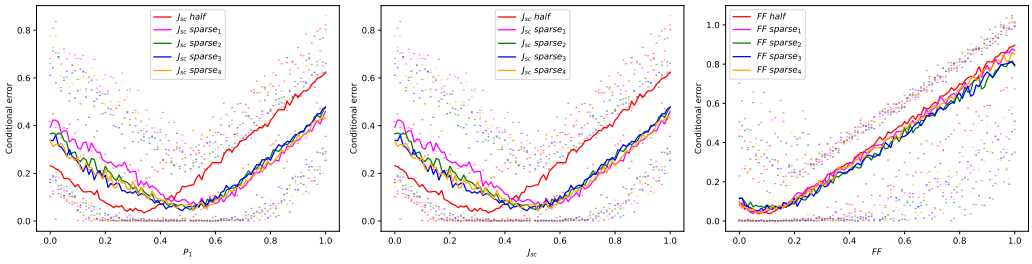
Figure 7: Target and optimal solutions for conditional optimization for P_1 objective with *half* dataset

5.4 Target conditioned optimization

5.4.1 Conditional generation

First, we evaluate the effectiveness of conditioning of our DGM using target objective values by sampling 128 targetted microstructures each for 100 uniformly sampled objective values and computing L_1 error with the evaluated objective values. This helps us understand if target objective conditioning is effective at generating required microstructures. Median conditional errors from target objective conditioning are shown in Figure 8 for different DGMs using solid lines. The minimum and maximum errors are shown as points of the corresponding colors. Conditional errors for objectives P_1 and J_{sc}, shown in Figure 8a and Figure 8b respectively show that all sparse datasets behave similarly with respect to conditional errors. This shows similar trends as the unconditional generation case that the DGM has intrinsically filled the gaps in the dataset. Conditional errors for *half* datasets quickly start increasing once data availability ends as conditioning capability depends on the availability of data. On the other hand, the FF dataset is heavily biased and the conditioning accuracy is affected by this bias. As shown in Figure 8c, both *sparse* and *half* datasets yield similar conditional errors as conditional error quickly increases after the dense region of the dataset. Nevertheless, the median errors demonstrate that the conditioning from target objectives is highly effective, giving confidence in using this during optimization. These values also show that conditioning alone might not be enough to replace the optimization procedure as it is not possible to know the amount of error in *unseen*

regions of the objective space. However, the minimum errors from these plots show that a localized search, as a fine-tuning step can yield accurate designs as targeted by the conditioning.



(a) P1 (b) J_{sc} (c) FF

Figure 8: Conditional errors for the trained DMs

5.4.2 Target conditional optimization

The effect of this target conditioning on the optimization procedure can be explained by comparing the trajectory with unconditional optimization. We report the objective value at the beginning (generation 1) and end of optimization. In the first generation, decision variables (z_{sem} in this case) are sampled uniformly from the given variable ranges. A low starting objective value implies that the target objective conditioning helps in starting optimization at a near-optimal solution. We observe in Table 2 that in 10 out of 15 problems (shown in bold, based on the Wilcoxon rank-sum test), target-conditioned optimization starts at a better objective value than the unconditional case. This shows that target-conditioning is effective at reducing the search space for optimization and the performed optimization acts as merely a local search. We also see this trend in Figure 5a where target-conditioned optimization (dotted lines) starts at a much lower value than unconditional case (solid lines). However, the final optimal solution found by target-conditional optimization might not be as good as the unconditional case as the reduced search space might not contain the optimum. Nevertheless, the possibility of target conditioning to reduce the size of search space and speed up optimization is a desirable feature of the proposed optimization framework. An example set of optimal microstructures are shown in Figure 7.

6 Conclusions and limitations

In this study, we proposed a multi-objective generative design framework based on diffusion autoencoders for microstructure-sensitive design and inverse problems. We rigorously evaluated the proposed framework based on two-phase microstructure dataset by framing various generative tasks, and single and multi-objective problems. We demonstrated that this framework provides a superior setting for generative design due to stability, ease of use, generative capability, and optimization landscape. While previous works have focused only on the quality of generated microstructure, we analyze the potency of this framework in various practical scenarios commonly encountered during generative design - like sparse and incomplete datasets and varying difficulties of optimization problems. We also demonstrated that this framework provides possibilities to generate multiple variants of a candidate design by varying the stochastic subcode of the diffusion autoencoder. This can be helpful in problems where small blemishes and artifacts can cause issues. Apart from these studies, we show that this framework can be modified to accommodate target objective conditioning in order to reduce the size of the search space and speed up the optimization procedure. One of the limitations of the work is that a simple conditioning procedure was used in this study and more attention needs to be devoted in the future to improving the accuracy of conditioning. Another limitation of this framework is the need for thousands of initial training images which might not be available in practical problems. A future study focusing on limited data needs to be performed. An important future direction is to use this framework for other problem classes and data structures, like graphs (molecular design) and meshes (finite element method based simulations). This framework can also be further extended to constrained problems, either as a constraint for optimization or generation.

Acknowledgments and Disclosure of Funding

We thank Ford Motor Company for funding this study.

References

[1] Ramin Noruzi, Sambit Ghadai, Onur Rauf Bingol, Adarsh Krishnamurthy, and Baskar Ganapathysubramanian. Nurbs-based microstructure design for organic photovoltaics. *Computer-Aided Design*, 118:102771, 2020.

[2] Pengfei Du, Adrian Zebrowski, Jaroslaw Zola, Baskar Ganapathysubramanian, and Olga Wodo. Microstructure design using graphs. *npj Computational Materials*, 4(1):50, 2018.

[3] Lyle Regenwetter, Amin Heyrani Nobari, and Faez Ahmed. Deep generative models in engineering design: A review. *Journal of Mechanical Design*, 144(7):071704, 2022.

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

[5] Zijiang Yang, Xiaolin Li, L Catherine Brinson, Alok N Choudhary, Wei Chen, and Ankit Agrawal. Microstructural materials design via deep adversarial learning methodology. *Journal of Mechanical Design*, 140(11), 2018.

[6] Daria Fokina, Ekaterina Muravleva, George Ovchinnikov, and Ivan Oseledets. Microstructure synthesis using style-based generative adversarial networks. *Physical Review E*, 101(4):043308, 2020.

[7] Hongrui Chen and Xingchen Liu. Geometry enhanced generative adversarial networks for random heterogeneous material representation. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 85383, page V03AT03A020. American Society of Mechanical Engineers, 2021.

[8] Ren Kai Tan, Nevin L Zhang, and Wenjing Ye. A deep learning–based method for the design of microstructural materials. *Structural and Multidisciplinary Optimization*, 61:1417–1438, 2020.

[9] Guillaume Lambard, Kazuhiko Yamazaki, and Masahiko Demura. Generation of highly realistic microstructural images of alloys from limited data with a style-based generative adversarial network. *Scientific Reports*, 13(1):566, 2023.

[10] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 2022.

[11] François Mazé and Faez Ahmed. Diffusion models beat gans on topology optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), Washington, DC*, 2023.

[12] Giorgio Giannone and Faez Ahmed. Diffusing the optimal topology: A generative optimization approach. *arXiv preprint arXiv:2303.09760*, 2023.

[13] Ethan Herron, Xian Yeow Lee, Aditya Balu, Balaji Sesha Sarath Pokuri, Baskar Ganapathysubramanian, Soumik Sarkar, and Adarsh Krishnamurthy. Generative design of material microstructures for organic solar cells using diffusion models. In *AI for Accelerated Materials Design NeurIPS 2022 Workshop*, 2022.

[14] Christian Düreth, Paul Seibert, Dennis Rücker, Stephanie Handford, Markus Kästner, and Maik Gude. Conditional diffusion-based microstructure reconstruction. *Materials Today Communications*, 35:105608, June 2023.

[15] Devesh Shah, Anirudh Suresh, Alemayehu Admasu, Devesh Upadhyay, and Kalyanmoy Deb. A survey on evaluation metrics for synthetic material micro-structure images from generative models. *arXiv preprint arXiv:2211.09727*, 2022.

[16] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10619–10629, 2022.

[17] K. Deb. *Multi-Objective Optimization Using Evolutionary Algorithms*. Wiley, Chichester, UK, 2001.

[18] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2):182–197, 2002.

[19] Conner Sharpe and Carolyn Conner Seepersad. Topology design with conditional generative adversarial networks. In *International design engineering technical conferences and computers and information in engineering conference*, volume 59186, page V02AT03A062. American Society of Mechanical Engineers, 2019.

[20] Zhenguo Nie, Tong Lin, Haoliang Jiang, and Levent Burak Kara. Topologygan: Topology optimization using generative adversarial networks based on physical fields over the initial domain. *Journal of Mechanical Design*, 143(3), 2021.

[21] Wei Chen and Mark Fuge. B\'eziergan: Automatic generation of smooth curves from interpretable low-dimensional parameters. *arXiv preprint arXiv:1808.08871*, 2018.

[22] Emre Yilmaz and Brian German. Conditional generative adversarial network framework for airfoil inverse design. In *AIAA aviation 2020 forum*, page 3185, 2020.

[23] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. arxiv e-prints. *arXiv preprint arXiv:1812.04948*, 2018.

[24] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020.

[25] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017.

[26] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

[27] Balaji Sesha Sarath Pokuri, Viraj Shah, Ameya Joshi, Chinmay Hegde, Soumik Sarkar, and Baskar Ganapathysubramanian. Binary 2d morphologies of polymer phase separation, February 2019.

[28] John W Cahn and John E Hilliard. Free energy of a nonuniform system. I. interfacial free energy. *The Journal of chemical physics*, 28(2):258–267, 1958.

[29] Balaji Sesha Sarath Pokuri, Sambuddha Ghosal, Apurva Kokate, Soumik Sarkar, and Baskar Ganapathysubramanian. Interpretable deep learning for guided microstructure-property explorations in photovoltaics. *npj Computational Materials*, 5(1):95, 2019.

[30] Hari K Kodali and Baskar Ganapathysubramanian. Computer simulation of heterogeneous polymer photovoltaic devices. *Modelling and Simulation in Materials Science and Engineering*, 20(3):035015, 2012.

[31] Manjeevan Seera, Choo Jun Tan, Kok-Keong Chong, and Chee Peng Lim. Performance analyses of various commercial photovoltaic modules based on local spectral irradiances in malaysia using genetic algorithm. *Energy*, 223:120009, 2021.

[32] J. Blank and K. Deb. pymoo: Multi-objective optimization in python. *IEEE Access*, 8:89497–89509, 2020.