Supplementary Material for

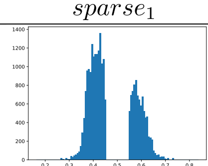
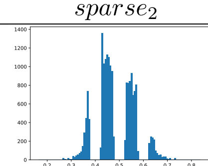
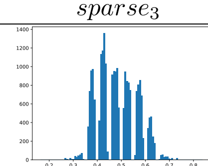
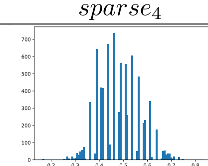
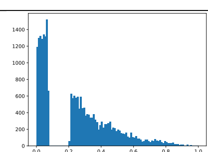
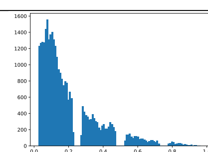
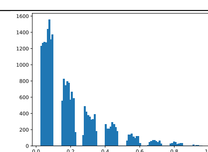
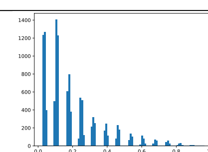
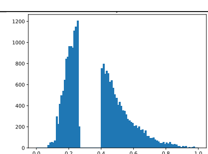
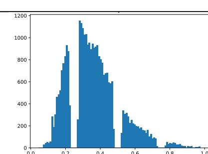
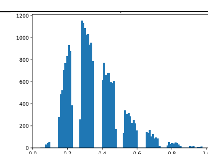
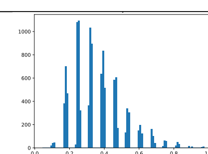
Multi-objective Evolutionary Design of Microstructures using Diffusion Autoencoders

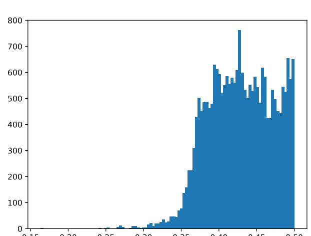
Anirudh Suresh[1] **Devesh Shah**[2] **Alemayehu Admasu**[2]
Devesh Upadhyay[2] **Kalyanmoy Deb**[1]
[1]Michigan State University [2] Ford Motor Company
{suresha2,kdeb}@msu.edu

1 Modified datasets

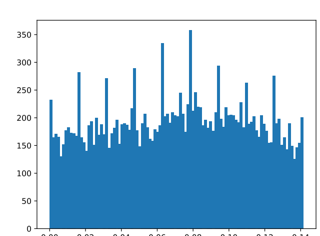
The modified datasets used for evaluating generative and optimization capabilities are shown in Table 1 and Figure 1. The amount of data in these modified datasets is reported in Table 2.

Table 1: Distribution of modified sparse datasets

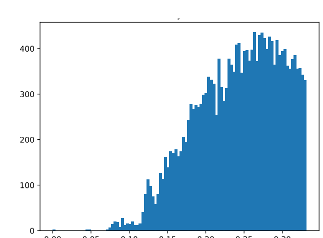
	$sparse_1$	$sparse_2$	$sparse_3$	$sparse_4$
P_1				
FF				
J_{sc}				



(a) P_1



(b) FF



(c) J_{sc}

Figure 1: Distribution of half datasets

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

Table 2: Amount of data in the modified datasets

Objective	Half	$sparse_1$	$sparse_2$	$sparse_3$	$sparse_4$
P_1	22712	23641	17828	22839	7735
FF	19296	23640	31380	23076	11773
J_{sc}	19291	23641	30162	23203	11644

2 DiffAE architecture details

All deep learning models are developed using Pytorch [1] and Pytorch Lightning [2]. DiffAE implementation is based on the original implementation [1]. Most of the architectural details and hyperparameters for training DiffAE models, as shown in Table 3 are based on the original paper [3].

Table 3: Network architecture and hyperparameters for DiffAE DGM

Parameter	**Unconditional**	**Conditional**
Batch size	64	64
Base channels	128	128
Channel multipliers	[1, 1, 2, 3, 4]	[1, 1, 2, 3, 4]
Attention resolution	16	16
Images trained	40M	40M
encoder attn res	16	16
encoder ch mult	[1, 1, 2, 3, 4, 4]	[1, 1, 2, 3, 4, 4]
z_{sem} size	128	112 + 16
Conditioning size	-	16
β scheduler	Linear	Linear
Learning rate	1e-4	1e-4
Optimizer	Adam	Adam
Training T	1000	1000
Diffusion loss	MSE with noise prediction	MSE with noise prediction

3 Surrogate model

A ResNet18 based surrogate model was trained with two regression outputs - FF and J_{sc}. The hyperparameters are shown below in Table 4.

Table 4: Architecture and hyperparameters for the surrogate models

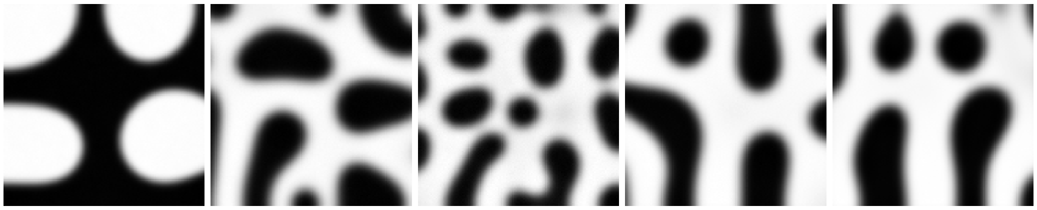
Parameter	Value
Architecture	ResNet18
Resolution	128
Batch size	128
Learning rate	1e-3
Lr schedule	Cosine annealing
Epochs	200

4 Results

In this section, we present some optimal microstructures generated during evolutionary optimization.

Figure 7 shows median progress of hypervolume during multi-objective optimization using NSGA-II. The hypervolume shows continuous improvement and convergence.

[1]https://github.com/phizaz/diffae

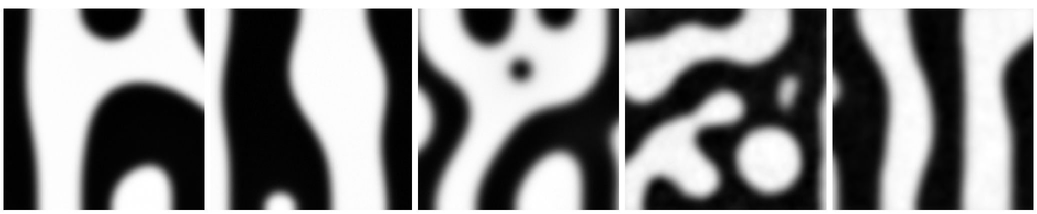


(a) 0.5 ($sparse_1$) (b) 0.55 ($half$) (c) 0.6 ($half$) (d) 0.65 ($half$) (e) 0.7 ($half$)

Figure 2: Targets and optimal solutions for single objective GA for P_1 objectives

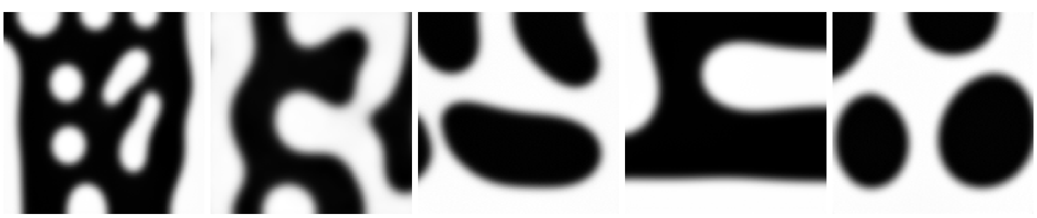
(a) 0.1415($sparse_1$) (b) 0.2 ($Half$) (c) 0.25($Half$) (d) $F0.3$($Half$) (e) 0.35($Half$)

Figure 3: Targets and optimal solutions for single objective GA for FF objectives



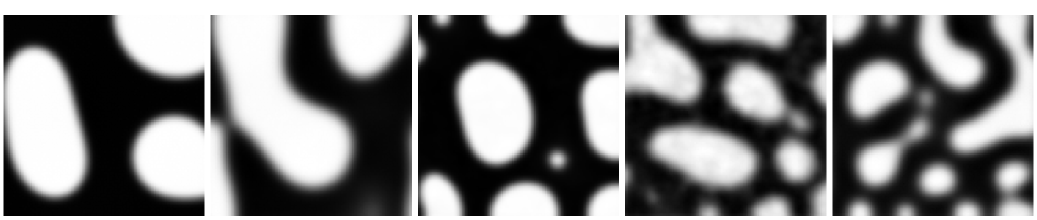
(a) 0.3314($sparse_1$) (b) 0.4($Half$) (c) 0.45($Half$) (d) 0.5($Half$) (e) 0.55($Half$)

Figure 4: Targets and optimal solutions for single objective GA for J_{sc} objectives



(a) 0.1415($sparse_1$) (b) 0.2 ($Half$) (c) 0.25 ($Half$) (d) 0.3 ($Half$) (e) 0.35 ($Half$)

Figure 5: Target and optimal solutions for conditional single objective GA for FF objectives



(a) 0.3314($sparse_1$) (b) 0.4 ($Half$) (c) 0.45 ($Half$) (d) 0.5 ($Half$) (e) 0.55 ($Half$)

Figure 6: Optimal solutions for conditional single objective GA for J_{sc} objectives

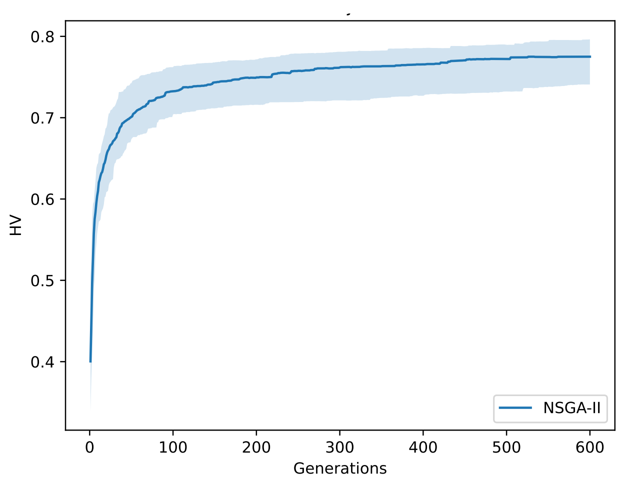


Figure 7: Hypervolume during multi-objective optimization

References

[1] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[2] William Falcon and The PyTorch Lightning team. PyTorch Lightning, March 2019.

[3] 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.