

Cite this: *Digital Discovery*, 2025, 4, 3175

3D multiphase heterogeneous microstructure generation using conditional latent diffusion models

Nirmal Baishnab, Ethan Herron, Aditya Balu, Soumik Sarkar, Adarsh Krishnamurthy* and Baskar Ganapathysubramanian *

The ability to generate 3D multiphase microstructures on-demand with targeted attributes can greatly accelerate the design of advanced materials. Here, we present a conditional latent diffusion model (LDM) framework that rapidly synthesizes high-fidelity 3D multiphase microstructures tailored to user specifications. Using this approach, we generate diverse two-phase and three-phase microstructures at high resolution (volumes of $128 \times 128 \times 64$ voxels, representing $>10^6$ voxels each) within seconds, overcoming the scalability and time limitations of traditional simulation-based methods. Key design features, such as desired volume fractions and tortuosities, are incorporated as controllable inputs to guide the generative process, ensuring that the output structures meet prescribed statistical and topological targets. Moreover, the framework predicts corresponding manufacturing (processing) parameters for each generated microstructure, helping to bridge the gap between digital microstructure design and experimental fabrication. While demonstrated on organic photovoltaic (OPV) active-layer morphologies, the flexible architecture of our approach makes it readily adaptable to other material systems and microstructure datasets. By combining computational efficiency, adaptability, and experimental relevance, this framework addresses major limitations of existing methods and offers a powerful tool for accelerated materials discovery.

Received 19th April 2025
Accepted 3rd September 2025

DOI: 10.1039/d5dd00159e

rsc.li/digitaldiscovery

1 Introduction

Understanding and controlling a material's microstructure is critical for optimizing its properties and performance. In materials science, the mapping between structure and property is a foundational concept, with microstructural features often serving as primary drivers of a material's physical characteristics and behavior.^{1–4} However, directly observing or reconstructing 3D microstructures through experiments is expensive and technically challenging, making it difficult to explore processing–structure–property relationships at scale.^{5–8} Consequently, there is a strong motivation to develop computational methods for generating realistic microstructures. The ability to produce statistically representative microstructure samples on-demand would greatly aid in virtual testing, microstructure-sensitive property prediction, and computational materials design.

Various approaches have been explored for microstructure generation.^{9,10} Classical statistical methods, such as Markov random fields,¹¹ Gaussian random fields,¹² and descriptor-based reconstructions,^{13,14} can produce microstructures that match certain target statistics. While these methods have

proven useful, they suffer from important limitations. In general, statistical models are computationally intensive and do not scale well to generating large 3D volumes or numerous samples. They often rely on strict assumptions (*e.g.* stationarity or isotropy of features) and tailored mathematical descriptors, which limits their flexibility and generalizability to different materials or complex structures. Adapting such models to incorporate new microstructural constraints or application-specific objectives is non-trivial and typically requires substantial rederivation or optimization changes. These challenges highlight the need for a more flexible, data-driven generative framework for microstructures.

Recently, deep generative models have shown great promise in capturing complex microstructural features from data.^{15,16} Approaches like variational autoencoders (VAEs),¹⁷ generative adversarial networks (GANs),¹⁸ and diffusion models (DMs)¹⁹ have been applied to microstructure generation tasks. VAEs can learn low-dimensional representations of microstructures but often produce blurry outputs that lack sharp detail.²⁰ GAN-based models have succeeded in generating 3D microstructures with improved visual fidelity,^{21–23} but they do not allow user control over generated structures and are notorious for training instabilities.²⁴ Moreover, GANs and similar networks can be computationally demanding for 3D data, sometimes requiring extensive resources for training and generation.

Iowa State University, Ames, IA, USA. E-mail: adarsh@iastate.edu; baskar@iastate.edu

Diffusion models offer even higher output quality, often surpassing GANs, but their iterative sampling process makes inference slow and resource-intensive.²⁵ At this time, no prior generative approach has simultaneously provided high fidelity, user controllability, and computational efficiency for 3D microstructure generation.

Latent diffusion models (LDMs) have emerged as a compelling solution to address these gaps.^{26,27} LDMs combine the strengths of VAEs and DMs by operating in a compressed latent space to dramatically reduce computational costs while preserving the ability to generate high-quality, diverse microstructures. This latent-space approach yields orders-of-magnitude speed-ups over conventional pixel-space diffusion models. Importantly, LDM architectures naturally support conditioning mechanisms that enable users to steer generation towards desired attributes. They also exhibit more stable training dynamics and avoid mode collapse, yielding a broader variety of outputs compared to GANs.^{28–31} These advantages make LDMs well-suited for fast and controllable 3D microstructure synthesis.

To date, applying diffusion-based generative models to microstructure design has predominantly focused on unconditional generation.^{32–34} In our prior work, Herron *et al.*³⁵ applied a diffusion model to 2D organic solar cell microstructures without enabling user-specified target features. While recent advances^{36,37} have begun exploring conditional generative approaches to microstructure reconstruction and design, these have typically not integrated predictions of corresponding manufacturing parameters. Our current work introduces a conditional latent diffusion modeling (LDM) framework that not only allows user-defined control over critical microstructural descriptors but also uniquely predicts manufacturing parameters likely to produce such microstructures experimentally. This two-fold capability addresses key challenges in computational materials design:^{38,39} not only can we generate microstructures with tailored properties, but we can also provide insight into how to manufacture them – thereby tackling the oft-cited “manufacturability gap” in microstructure design.

We demonstrate the framework using organic photovoltaic (OPV) active-layer microstructures as a representative example. OPV active layers typically consist of a donor material and an acceptor material, forming a complex two-phase (or three-phase with a mixed phase) morphology.⁴⁰ Two microstructural descriptors are particularly crucial for OPV performance: the donor (acceptor) phase volume fraction and the tortuosity of the percolating pathways.^{41,42} The volume fraction (the ratio of donor to acceptor material in the blend) directly influences the balance between charge generation and transport, while tortuosity reflects the complexity of pathways that charge carriers must navigate to reach the electrodes. By conditioning on these properties in the LDM, we can generate microstructures that meet specific targets (*e.g.* a desired donor volume fraction and phase connectivity) known to optimize OPV efficiency. We quantify volume fraction and tortuosity for each generated sample using established computational techniques.⁴³

The key contributions of this work include: (1) scalable high-resolution 3D microstructure generation: leveraging an LDM, we

rapidly produce diverse multiphase 3D microstructures (including two-phase and three-phase examples) at a resolution of $128 \times 128 \times 64$ voxels (over one million voxels each), which is orders of magnitude larger than those demonstrated in prior studies. Our approach generates these 3D microstructures in seconds per sample. (2) Conditional generation with user-defined features: our framework introduces controllability to microstructure synthesis by allowing users to specify target volume fractions and tortuosities; the LDM then generates microstructures that faithfully realize these input parameters, ensuring the output matches desired structural characteristics. (3) Linking microstructure to manufacturing: we integrate a predictive module that outputs relevant processing parameters (*e.g.* annealing or fabrication conditions) corresponding to each generated microstructure, facilitating a direct connection between the digital microstructure design and its experimental realization. These advances collectively overcome the scalability, controllability, and manufacturability limitations of existing methods. By enabling fast generation of application-specific microstructures along with guidance for their fabrication, our conditional LDM framework illustrates the promise of AI-driven approaches in computational materials science and microstructure design.

2 Results and discussion

To demonstrate our framework's capabilities, we evaluated its performance using both synthetic microstructures generated *via* physics-based simulations (Cahn–Hilliard equation) and experimentally obtained (*via* tomography) organic photovoltaic (OPV) morphologies. We will refer to these datasets as the CH dataset and the experimental dataset, respectively. The results illustrate the advantages of our conditional latent diffusion modeling (LDM) approach in generating diverse, high-quality microstructures efficiently and with precision.

Our proposed generative modeling framework, schematically illustrated in Fig. 10, consists of three sequentially trained modules: a Variational Autoencoder (VAE), a Feature Predictor (FP), and the Latent Diffusion Model (LDM). Initially, the VAE compresses complex, high-dimensional 3D microstructures into compact latent representations, drastically reducing computational complexity. The FP network subsequently predicts relevant microstructural features (*e.g.*, volume fractions and tortuosities) and manufacturing parameters directly from these latent representations. Finally, the conditional LDM leverages these predictions to generate realistic 3D microstructures, guided explicitly by user-specified conditions.

In the following sub-sections, we detail our evaluation of the framework's generative capabilities, including the quality and diversity of generated microstructures, the effectiveness of conditional sampling for targeted microstructure design, and the model's unique capacity to predict experimental manufacturing parameters.

2.1 Sampling quality

Fig. 1 shows representative examples of microstructures generated by our Latent Diffusion Models (LDMs), separately



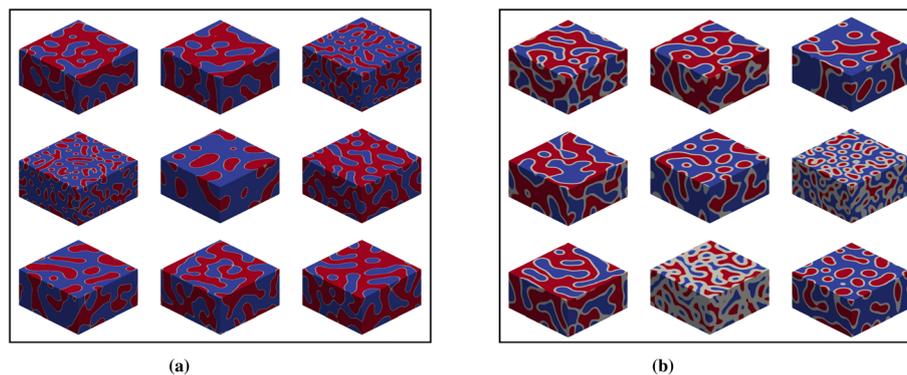


Fig. 1 Samples from LDMs trained on (a) two phase and (b) three phase microstructures.

trained for the CH dataset's two-phase and three-phase systems. In the two-phase microstructures, the blue domains represent a donor phase (denoted phase A), and the red domains represent an acceptor phase (denoted phase B), corresponding to typical organic photovoltaic (OPV) active-layer morphologies. For the three-phase microstructures, an additional gray phase delineates a mixed region, that typically exists as an interfacial region between donor and acceptor phases.

Each generated microstructure spans a volume of $128 \times 128 \times 64$ voxels, corresponding to over one million voxels (1 048 576), allowing detailed resolution of intricate morphological features. Importantly, our LDM framework achieves this generation within approximately 0.5 seconds per microstructure using an NVIDIA A100 GPU, significantly outperforming traditional physics-based simulation methods, which typically require hours or days of computation for similar-sized volumes.^{44–46} The transition from two-phase to three-phase systems maintains high quality and fidelity, demonstrating the flexibility and scalability of our framework. Without any modification to the core architecture, retraining on a three-phase dataset successfully generated microstructures exhibiting smaller domains and more complex, finely detailed features. This ease of adaptability underscores the potential for further extension of our approach to accommodate additional phases.

2.2 Conditional sampling

In this work, conditional sampling refers to the approach of providing the generative model with additional information—termed a conditioning vector—to guide the synthesis of microstructures toward specific, user-defined characteristics. We implemented this conditional generation by embedding the conditioning vector directly into the latent diffusion model (LDM), allowing precise control over the structural features of the generated microstructures. Specifically, the LDM architecture incorporates the conditioning vector into the embedding layers of the U-Net backbone, facilitating effective guidance during the diffusion process.

The LDM is conditioned on two crucial microstructural descriptors relevant to organic photovoltaics: the volume fractions and tortuosities of the phases (A, B, and the mixed phase). However, our flexible conditioning framework is easily extensible to other relevant morphological descriptors, depending on the application requirements (see additional examples provided in the SI Results). Fig. 2 illustrates representative examples of conditionally generated microstructures, clearly demonstrating the effectiveness of the model in synthesizing morphologies tailored to user-specified volume fractions and tortuosities.

To evaluate the model's ability to generate conditional outputs, we created 3200 microstructures with different targeted volume fractions and tortuosity values. We systematically

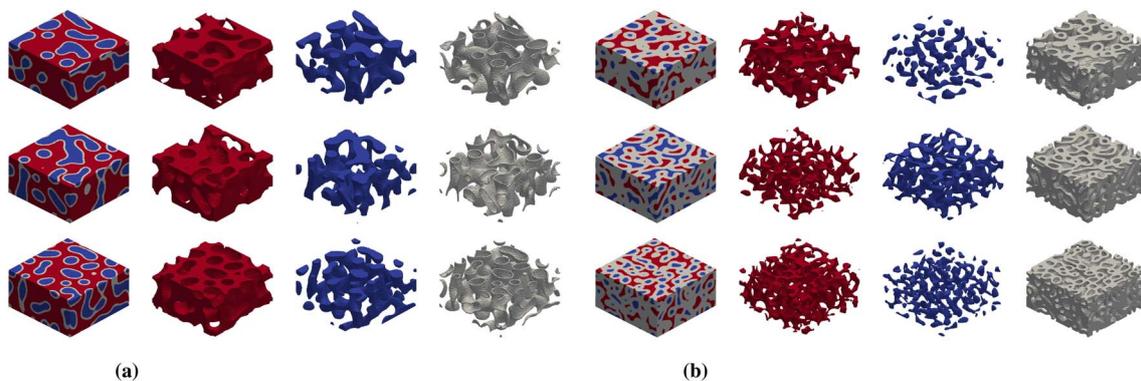


Fig. 2 Conditional microstructure generation: sample microstructures from user inputs – (a) predominant phase A, and (b) predominant phase mixed. First column shows the total microstructure. Second, third and fourth columns show the thresholded versions of the phase A, phase B and mixed components, respectively.



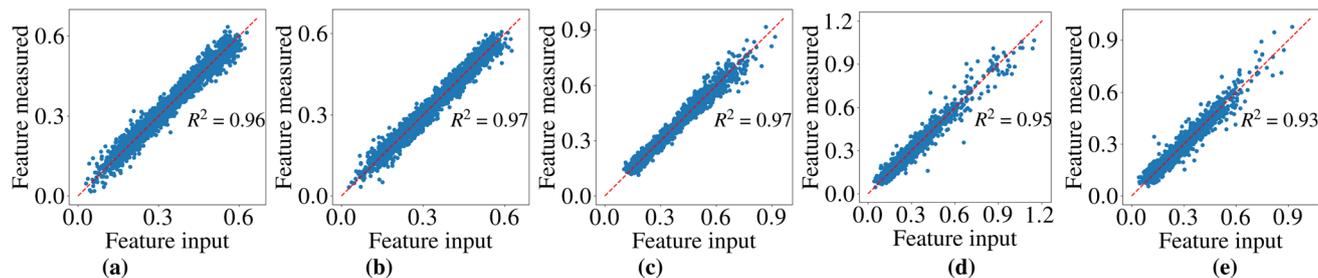


Fig. 3 Statistical analysis of conditional microstructure generation: correlations between all features of interest, user inputs, and the corresponding features measured from generated microstructures. (a) Phase A volume fraction. (b) Phase B volume fraction. (c) Mixed volume fraction. (d) Tortuosity A. (e) Tortuosity B.

compared these microstructure attributes with the user-specified conditioning parameters, as depicted in Fig. 3. Our analysis reveals a high degree of accuracy in conditional generation, achieving Pearson correlation coefficients (R^2) of 0.93 or greater. This robust correlation underscores the LDM's effectiveness in adhering to precise user-defined constraints, thereby enabling targeted material design and optimization that surpasses prior methods in versatility and computational efficiency.^{22,23}

As with most data-driven generative frameworks, the proposed LDM model learns and reproduces the joint distribution of microstructural features in the training data. In physics-based datasets such as our CH dataset, certain features (e.g., volume fraction and tortuosity) naturally exhibit correlations due to underlying physical constraints. Consequently, the generative model tends to reflect these correlations and may struggle to generate feature combinations that are poorly represented or absent in the training dataset. However, the framework remains flexible and, in principle, capable of learning a broader range of feature combinations if provided with sufficiently diverse and decorrelated training data. As the diversity and coverage of the training dataset increase, the model's ability to generate microstructures with uncommon or more complex feature relationships is expected to improve accordingly. We conducted experiments using more conditioning parameters to assess the framework's capacity for higher-dimensional conditioning. Appendix Fig. 16 presents the results with seven conditioning parameters. Increasing the number of conditioning parameters introduces two challenges. First, the model must learn more complex and potentially correlated feature relationships. Second, as the dimensionality grows, the volume of the conditioning space expands rapidly, resulting in a sparser sampling of the parameter space, which in turn demands a larger and more diverse training dataset. Fig. 16 shows a decline in R^2 between the input features and the measured features of the generated microstructures, yet the model still maintains strong correlations. If the conditioning features are not fully independent but exhibit correlations, care must be taken to ensure that valid and physically meaningful combinations are used during inference. In such scenarios, dimensionality reduction techniques (e.g., principal component analysis or other embedding methods) may be employed to

reduce the effective dimensionality of the conditioning space prior to model training.

Alternative conditional generative approaches for microstructure design have recently been reported. For example, Gao *et al.*³⁶ introduced a deep learning framework for multi-scale prediction of mechanical properties from microstructural features in polycrystalline materials, while Lee and Yun³⁷ developed a denoising diffusion-based method for generating three-dimensional anisotropic microstructures from two-dimensional micrographs. While these works incorporate conditional elements, they do not provide the combined capability of user-defined control over specific microstructural descriptors and simultaneous prediction of manufacturing parameters. Our conditional latent diffusion framework thus addresses a different design space—high-resolution descriptor-controlled generation.

2.3 Diversity and prediction of manufacturing parameters

We further assessed the LDM's capability to generate diverse microstructures from identical conditional inputs. Specifically, we sampled 3200 microstructures using consistent input parameters (volume fractions: 0.3 for phase A, 0.2 for the mixed phase; tortuosities for both phases: 0.3). The resulting microstructures, detailed in the SI, exhibit significant morphological diversity despite identical conditioning parameters. Fig. 4a illustrates the distributions of the extracted microstructural features, clearly aligning with the specified input values (indicated by vertical dotted lines). The strong alignment confirms that the LDM reliably generates diverse yet precisely targeted microstructures.

Moreover, Fig. 4b presents contour plots predicting the manufacturing parameters—the blend ratio, the interaction parameter (χ), and the annealing time (timesteps)—required for realizing these microstructures. Notably, the LDM framework identifies multiple feasible fabrication pathways: a combination of higher χ values with shorter annealing durations, or lower χ values with extended annealing periods. This data-driven insight aligns well with the known physical behavior of phase-separating systems described by the Cahn–Hilliard model, where increased interaction parameters accelerate phase separation, thereby requiring less annealing time, whereas lower interaction parameters necessitate longer annealing to achieve



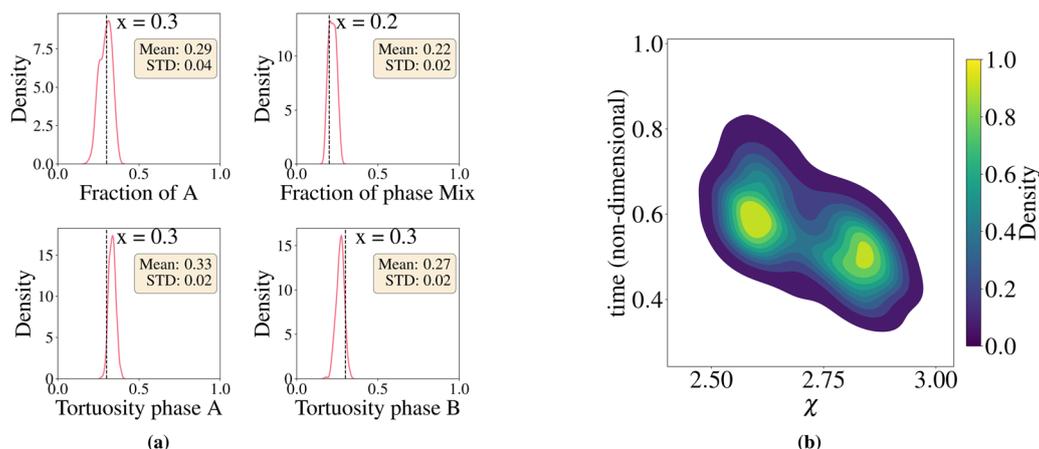


Fig. 4 Variety of microstructures generated by the LDM given identical user inputs. The model can also suggest the manufacturing conditions required to generate such microstructures. (a) Distribution of features measured from generated microstructures given specific conditional feature inputs. The vertical dotted black lines indicate the user inputs. (b) Contour plot of manufacturing parameters χ and timesteps for desired microstructure generation.

comparable morphologies. This pathway prediction capability illustrates the integration of computational design with experimental manufacturability, thus significantly advancing current microstructure design methodologies.^{22,23} Such an approach could be expanded to include other manufacturing parameters, making the model applicable across various material systems and manufacturing processes.^{22,23}

Although the training dataset includes time-dependent snapshots generated from Cahn–Hilliard simulations, the generative model itself operates solely on static 3D microstructures paired with their corresponding morphological descriptors. The time-dependent simulations are used primarily to provide a diverse and physically meaningful training set across a range of morphologies. The generative model remains agnostic to the physical dynamics or governing equations responsible for generating the dataset. This formulation enables flexible, descriptor-driven microstructure generation. Future work could explore the incorporation of additional physics-based constraints, such as mass conservation or dynamic evolution, for applications requiring dynamic modeling.

2.4 Experimental microstructures

We further demonstrated our framework's applicability using the experimental dataset comprising voxelized organic photovoltaic (OPV) morphologies from spin-cast P3HT:PCBM thin films, reconstructed through tomographic energy-filtered TEM^{47,48} (additional methodological details are provided in the Methods section).

Using the experimental dataset, we generated 1000 microstructures conditioned on user-specified inputs. Fig. 5 shows the correlation between the specified inputs and the corresponding measured features, with Pearson R^2 values of 0.89 for volume fraction, 0.86 for acceptor tortuosity, and 0.77 for donor tortuosity. These values are somewhat lower than those observed for the CH dataset; however, they remain reasonably strong given the characteristics of the experimental data. First, the experimental microstructures are lower in resolution but contain finer-scale features, which limits the ability of the latent diffusion model to capture the details. Second, the experimental dataset is less diverse: the subvolumes are extracted from only two larger tomographic samples, with overlapping subregions, resulting in a narrower sampling of the feature

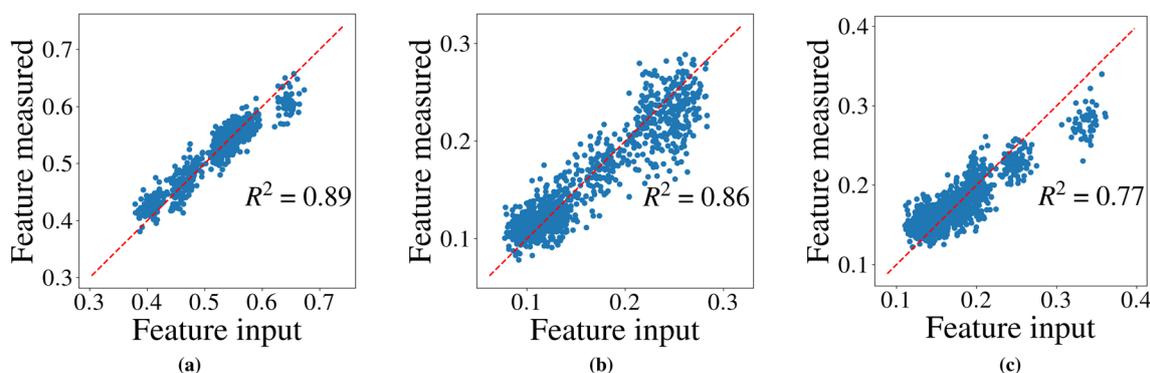


Fig. 5 Statistical analysis of conditional microstructure generation: correlations between all features of interest, user inputs, and the corresponding features measured from generated microstructures. (a) Donor volume fraction. (b) Tortuosity acceptor. (c) Tortuosity donor.



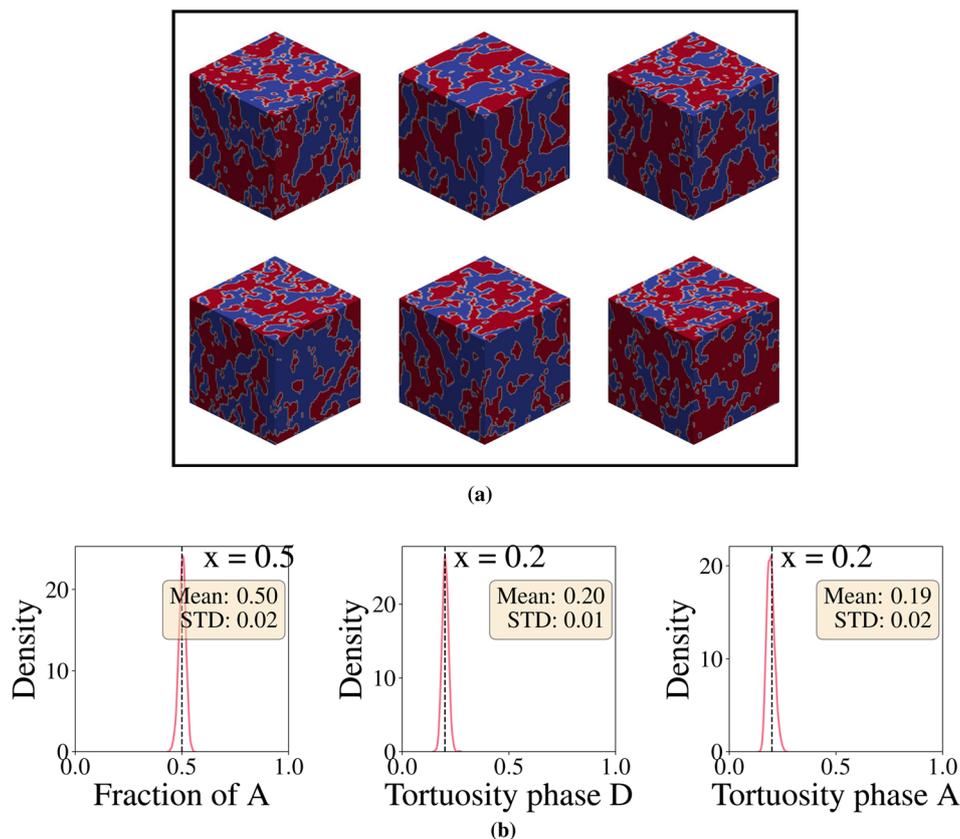


Fig. 6 Variety of microstructures generated by the LDM given identical user inputs. The model can also suggest the manufacturing conditions required to generate such microstructures. (a) Samples microstructures generated from same conditional feature inputs. (b) Distribution of features measured from generated microstructures given specific conditional feature inputs. The vertical dotted black lines indicate the user inputs.

space. These factors inherently constrain the achievable correlation between target features and generated structures. Nevertheless, the model captures the volume fraction with higher accuracy, as it is a simpler global descriptor. In contrast, tortuosity, a more localized and structurally complex feature, potentially requires better resolution and poses greater modeling challenges.

Additionally, Fig. 6a presents six representative microstructures generated from identical conditioning inputs (volume fraction: 0.5; donor and acceptor tortuosities: 0.2 each), illustrating notable morphological diversity. The kernel density estimation (KDE) plots shown in Fig. 6b confirm that the generated feature distributions are closely centered around the specified target values, with standard deviations of 0.02 or less, highlighting the precision and robustness of the conditional LDM in practical, experimental contexts.

2.5 Inference performance analysis

Table 1 summarizes both datasets' model sizes and parameter counts. The CH dataset contains microstructures with 4 times more voxels than the experimental dataset. However, the latent dimension is 2 times larger for the experimental dataset. We used a lower compression ratio to capture finer details in the experimental structures. In the CH dataset, the latent

representation has four channels, while the experimental dataset uses only one channel. During tuning, increasing the number of channels for the experimental dataset did not improve reconstruction loss. These choices were based on empirical hyperparameter tuning without a specific rule. Despite differences in image size and latent space, both models have comparable VAE and DDPM parameter counts, as the encoder-decoder architecture remains consistent mainly across datasets. The feature predictor has a smaller parameter count for the experimental dataset because it predicts fewer conditional and manufacturing parameters.

Table 1 Model configurations and sizes for both datasets

Model component	CH dataset	Experimental dataset
Input size	$128 \times 128 \times 64$	$64 \times 64 \times 64$
Latent dimension	$4 \times 8 \times 8 \times 4$	$1 \times 8 \times 8 \times 8$
Conditional parameters	4	3
Manufacturing parameters	3	0
VAE size (MB)	178.97	178.96
VAE parameters	46 916 781	46 912 836
DDPM size (MB)	575.55	575.23
DDPM parameters	150 871 044	150 786 561
Feature predictor size (MB)	2.51	0.63
Feature predictor parameters	657 927	164 611



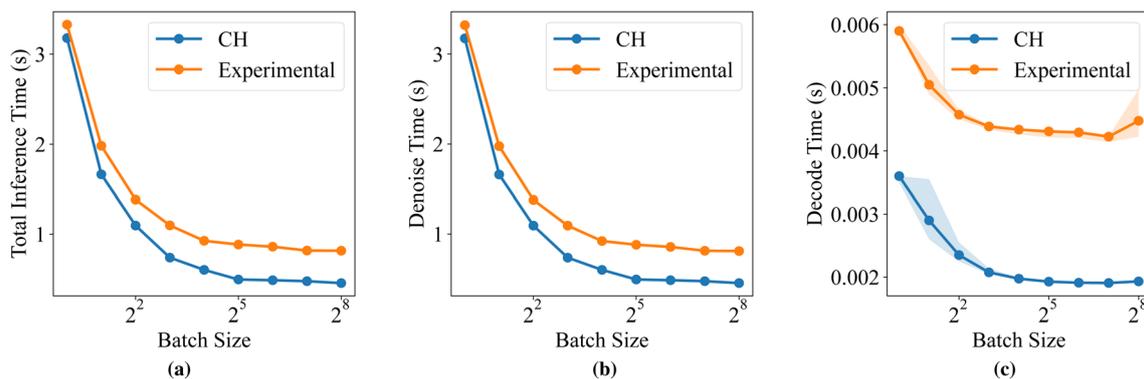


Fig. 7 Inference time per sample breakdown as a function of batch size for both the CH and experimental OPV datasets. Shaded regions indicate min–max variation across runs. (a) Total inference time. (b) Denoise time. (c) Decode time.

Fig. 7 presents a breakdown of inference performance for both datasets across varying batch sizes. For each dataset, we report the total inference time, along with a decomposition into denoising and decoding times. The model demonstrates parallel scalability up to a batch size of 32, beyond which the time per sample plateaus at approximately 0.5 s for the CH dataset and 0.8 s for the experimental dataset. Although the experimental dataset has a smaller total latent size, its latent representation has fewer channels and larger spatial dimensions per channel, which leads to less efficient parallelization. In contrast, the CH dataset, with more channels, better utilizes GPU parallelism at the kernel level. Across all configurations, denoising remains the dominant computational cost, while decoding contributes minimally. For example, at a batch size of 32, denoising takes over 200 times longer than decoding for both datasets. This behavior is consistent with diffusion models, where the denoising process involves iterative sampling—in our case, 1000 iterations per sample.

3 Conclusions

Conditional microstructure generation can be useful across different fields. For example, energy storage, biomedical devices, and additive manufacturing. We have presented a conditional latent diffusion framework capable of generating high-fidelity 3D multiphase microstructures conditioned on user-specified features. Our framework has been tested on both experimental and physics-based simulation datasets. It demonstrates strong control over key morphological descriptors, showing high correlations between the target and generated features. Currently, the three components of the framework (Variational Autoencoder, Feature Predictor, and Denoising Diffusion Probabilistic Model) are being trained sequentially. Future work includes streamlining and parallelizing the training process to reduce overall training time. We aim to incorporate parameter validation and automatic parameter selection for the conditional inputs in the inference pipeline. Future research will also explore applying this framework to other datasets across various domains.

4 Methodology

4.1 Training dataset

The computational dataset used in this project was synthesized from three-dimensional simulations of the Cahn–Hilliard equation, solved using the Finite Element Method (FEM). It comprises a wide range of phase separation scenarios, captured through simulations under varying conditions defined by two parameters: the initial volume fraction (ϕ) and the Flory–Huggins interaction parameter (χ). The Cahn–Hilliard equation represents a microstructure by modeling the spatial variation of two or three components. In our dataset, ϕ is varied systematically to explore a wide spectrum of initial mixture compositions, capturing the dynamics of phase separation. The interaction parameter, χ , is another key variable in the dataset. It quantifies the degree of affinity or aversion between the mixture's components. A higher χ value signifies a strong tendency towards phase separation due to energetically unfavorable interactions, while a lower value suggests better miscibility. By altering χ , we probe different interaction regimes, from weak to strong phase-separating tendencies. For each combination of ϕ and χ , the dataset captures over 400 time-stamped snapshots of a 3D Cahn–Hilliard simulation at $128 \times 128 \times 64$ resolution, providing a detailed temporal sequence of the phase separation process. There are 67 such time series, resulting in a total of over 26 800 3D microstructures. The dataset was divided into training and validation sets, with 80% of the data allocated to training and 20% to validation.

From these microstructures, we performed thresholding to obtain two-phase and three-phase representations. In the two-phase case, for example, voxels with values below 0.5 were assigned to one phase (0), while those above 0.5 were assigned to the other phase (1). The three-phase microstructures were generated by applying multi-level thresholding to the simulated continuous microstructure fields. Two threshold values were selected to partition the field into three distinct regions (donor, acceptor, and interface), each corresponding to one of the phases. There were no fixed threshold levels; the levels were adjusted based on the original microstructure to ensure that the interface did not become too thick compared to the donor and



acceptor phases. Based on the thresholded microstructures, morphological descriptors such as volume fractions and tortuosities were calculated. The volume fraction was computed as the ratio of the number of voxels belonging to a given phase to the total number of voxels in the microstructure. In the context of OPV, tortuosity is quantified as the fraction of phase-connected voxels exhibiting straight rising paths (*i.e.*, with a tortuosity of 1) to their respective electrodes.^{43,49} Specifically, donor tortuosity refers to the fraction of black voxels (donor phase) that are connected to the anode (top electrode or top edge of the microstructure) *via* straight rising paths, while acceptor tortuosity refers to the fraction of white voxels (acceptor phase) connected to the cathode (bottom electrode or bottom edge). Tortuosity is a critical microstructural descriptor in OPV because it captures the efficiency of charge transport pathways within the active layer. We used the graph-based tool GraSPI⁵⁰ to compute these descriptors. GraSPI provides a suite of microstructural descriptors that are particularly relevant to the analysis and performance evaluation of organic solar cells.

Fig. 8 shows snapshots from a single time series within the training dataset. The snapshots represent the temporal evolution of phase separation during the 3D simulation of the Cahn–Hilliard equation, illustrating the dynamic changes in microstructures over time. The Cahn–Hilliard model accounts for both thermodynamic forces and kinetic processes driving phase separation, providing insights into how processing conditions, such as annealing, influence the final morphology of the active layer. This understanding can aid to the optimization of material processing to improve organic solar cell (OSC) performance.^{51,52}

In addition to the computational dataset, we also utilized voxelized experimental OPV morphologies from spin-cast P3HT:PCBM thin films fabricated using two different solvents: chlorobenzene (CB) and dichlorobenzene (DCB). These morphologies were fabricated and reconstructed using tomographic energy-filtered TEM (see Heiber *et al.*,⁴⁷ Herzing *et al.*⁴⁸ for details). The imaging volume had approximate dimensions of $1\ \mu\text{m} \times 1\ \mu\text{m} \times 100\ \text{nm}$, with the EF-TEM-based

reconstruction achieving a voxel resolution of approximately 2.12 nm. The CB morphology is depicted in Fig. 9, where blue domains represent the electron-donating (donor) materials and red domains indicate the electron-accepting (acceptor) materials. The voxelized resolutions of the CB and DCB morphologies are $466 \times 465 \times 50$ and $478 \times 463 \times 60$, respectively. To generate a uniform dataset, we extracted cubic subvolumes spanning the full *z*-axis of each morphology and resized them to $64 \times 64 \times 64$ using nearest-neighbor interpolation. In the *x* and *y* directions, we used a step size of 4 voxels, resulting in over 10 500 cubic subvolumes of size $64 \times 64 \times 64$ from each of the two main morphologies. This process yielded a total of over 21 000 $64 \times 64 \times 64$ 3D microstructures. A similar subvolume extraction and segmentation strategy has been used in related 3D microstructure studies,⁵³ where high-resolution FIB-SEM images were segmented into voxelized phase maps for downstream model training. Similar to the synthetic dataset, this dataset was also divided into training and validation sets in the usual 80–20% split.



Fig. 9 Visualization of spin-cast P3HT:PCBM thin film, fabricated using chlorobenzene reconstructed using tomographic energy-filtered TEM. The main image shows the reconstructed 3D morphology, with blue and red domains representing the electron-donating (donor) and electron-accepting (acceptor) materials, respectively. The inset provides a zoomed-in view of a cubic subvolume extracted from the full morphology.

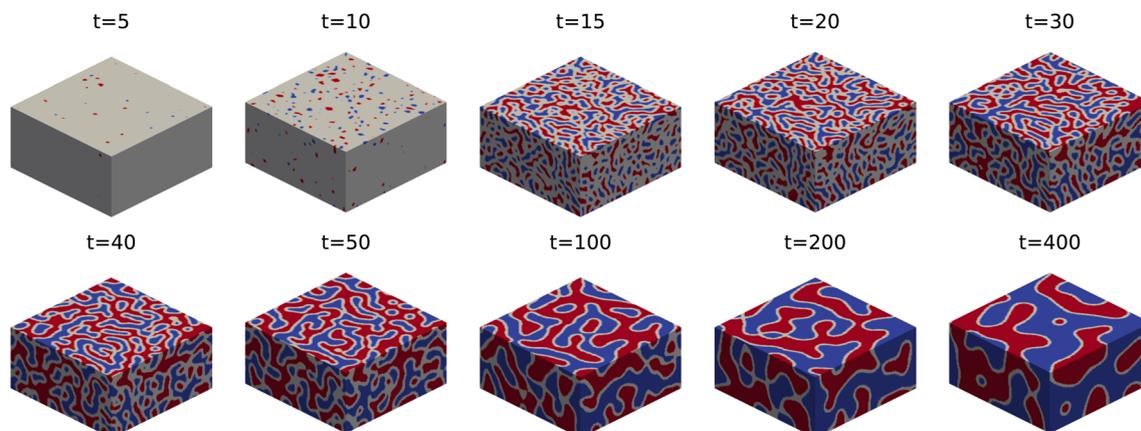


Fig. 8 A sequence of 10 snapshots from one time series out of 67 in the entire dataset, illustrating the evolution of phase separation in a 3D simulation of the Cahn–Hilliard equation.



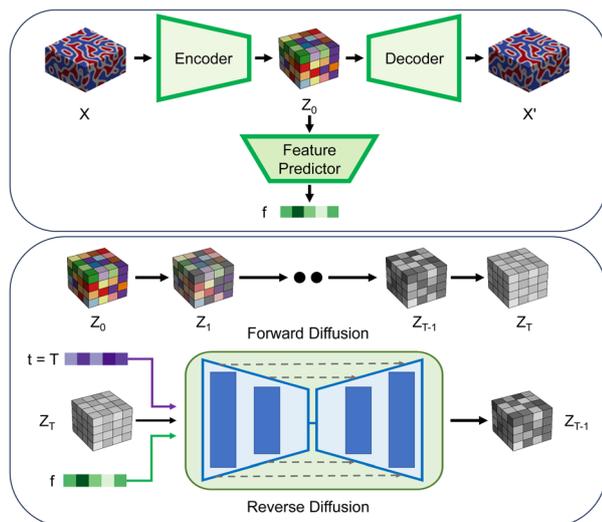


Fig. 10 Overview of the proposed LDM-based framework's three-step training process: VAE training and latent representation dataset creation, training of the FP, training of DM in the latent space.

4.2 Generative model architecture

The architecture of the training framework is provided in Fig. 10. The core of our generative framework is the LDM, which offers several advantages over traditional DMs. LDMs are superior in computational efficiency, memory usage, generation speed, and scalability.^{26,30} They excel in processing 3D data, operating in a lower-dimensional latent space that significantly reduces the computational load. This approach not only accelerates generation but also decreases memory requirements—crucial for handling complex 3D datasets. The reduced computational and memory demands allow for quicker iterations, making LDMs ideal for applications that require rapid prototyping or extensive simulations.

Additionally, the scalability of LDMs enables them to manage larger datasets and more complex microstructures without a proportional increase in resource consumption, unlike traditional DMs. This combination of factors renders LDMs a more efficient and practical choice for generating detailed 3D microstructures in a resource-conscious manner. Our LDM framework comprises three components: a VAE, a Feature Predictor (FP), and a DM, which are trained sequentially. The encoder and decoder of the VAE are trained simultaneously to obtain the latent space from which the FP is trained. Once the VAE and FP are trained, we train the DM using the latent space and the predicted features.

4.2.1 Variational autoencoder. Contrary to classic autoencoders that transform an input x directly into a latent representation z , VAEs convert x into a probability distribution.¹⁷ In VAEs, the encoder doesn't predict a single point but instead determines the mean and variance of this distribution. The latent variable z is then derived from this distribution. This is done by initially sampling from a standard normal Gaussian distribution, then scaling this sample with the predicted variance, and finally, adding the predicted mean to this scaled value.

To generate a sample z from the latent space, the VAE uses a random sample ε drawn from a standard normal distribution:

$$z = \mu_\phi(x) + \sigma_\phi(x) \odot \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \mathbf{I}) \quad (1)$$

where \odot denotes element-wise multiplication. The encoder maps the input x to two parameters in the latent space – the mean μ and the log-variance (log-var):

$$q_\phi(z|x) = \mathcal{N}(z; \mu_\phi(x), \exp(\log - \text{var}_\phi(x))) \quad (2)$$

The decoder maps the latent representation z back to the input space:

$$p_\theta(x|z) = \mathcal{N}(x; \mu_\theta(z), \exp(\log - \text{var}_\theta(z))) \quad (3)$$

The loss function in VAEs consists of two terms, the reconstruction loss and the KL divergence:

$$\mathcal{L}(\theta, \phi; x) = -\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] + \text{KL}(q_\phi(z|x)||p(z)) \quad (4)$$

This function balances the accuracy of reconstruction with the regularization of the latent space.

The VAE is the entry point for our architecture. The VAE employed in this work consists of an encoder-decoder structure with residual blocks for feature extraction and reconstruction. The encoder comprises five 3D convolutional layers, each followed by Instance Normalization and a residual block to capture spatial dependencies in the input data. The latent space is parameterized by a mean ('mu') and log-variance ('logvar'), both of which are obtained through additional 3D convolutional layers. The decoder mirrors the encoder's structure, using transposed convolutions to upsample the latent space back to the original input dimensions with residual blocks and Instance Normalization for stable training. A final Sigmoid activation is applied to the output to generate the reconstructed data. Once the VAE is trained, we use its encoder to compress microstructures with over a million voxels into a compact encoded representation of size 1024 ($4 \times 8 \times 8 \times 4$), while for experimental VAE inputs of $64 \times 64 \times 64$ (over 262 K voxels), the output is further reduced to 512 ($1 \times 8 \times 8 \times 8$). This reduced-dimensional latent space, distinguished by its efficiently learned data distribution, facilitates more efficient and stable diffusion processes.

4.2.2 Feature predictor. The feature predictor is a fully connected neural network designed to predict specific microstructural and manufacturing features based on encoded representations of 3D morphological data. The model architecture includes an input layer, two hidden layers, and an output layer. The input layer receives a flattened latent representation of size 1024, generated by a pretrained VAE. This representation is then processed through two hidden layers, each reducing the data dimensionality while applying instance normalization and dropout (dropout = 0.1) to prevent overfitting. The final output layer maps the processed data to the desired number of features, which correspond to the predicted manufacturing and morphological characteristics.



4.2.3 Diffusion model. DMs consist of two main stages: the forward diffusion and the backward diffusion. In the forward diffusion stage, Gaussian noise is repeatedly added to a data sample drawn from a specific target distribution. This process is performed multiple times, resulting in a series of samples that become increasingly noisy compared to the original data. In this work, we use the original Denoising Diffusion Probabilistic Model (DDPM)⁵⁴ formulation. In DDPMs the forward process is described by the Markov chain:

$$q(x_t|x_{t-1}) = \mathcal{N}\left(x_t; \sqrt{1 - \beta_t}x_{t-1}, \beta_t\mathbf{I}\right) \quad (5)$$

where x_0 is the initial sample from the target distribution $q(x)$, and the variance schedule is defined as $\{\beta_t \in (0,1)\}_{t=1}^T$. Conversely, the backward diffusion stage aims to iteratively eliminate the noise introduced in the forward stage, represented as $q(x_{t-1}|x_t)$. Direct sampling from $q(x_{t-1}|x_t)$ is not possible because that would require the complete knowledge of the distribution. Therefore, the model uses a neural network $G_\theta(x_{t-1}|x_t)$, parameterized by G and θ , to approximate these conditional probabilities. The network, refined through gradient-based optimization, aims to replicate the random Gaussian noise used in the forward diffusion process for transforming the original sample into a noisy version x_t at a particular timestep. The objective function is expressed as:

$$\|z - G_\theta(x_t, t)\|^2 = \mathbb{E}_{t,x_0,z} \left[\|z - G_\theta(\sqrt{\alpha_t}x_0 + \sqrt{1 - \alpha_t}z, t)\|^2 \right] \quad (6)$$

Here, $\alpha_t = 1 - \beta_t$, $\bar{\alpha}_t = \prod_{s=1}^t \alpha_s$, and $z \sim \mathcal{N}(0, \mathbf{I})$.

The neural network's primary role in a DM is to learn the inverse of the noise addition process. By systematically removing the noise added during the forward diffusion process, the network reconstructs the original data from its noisier versions. This process enables the generation of new, high-quality samples from completely random Gaussian noise. More concretely, once the DM has been trained, we can generate a new latent sample by starting with random Gaussian noise $z \sim \mathcal{N}(0, \mathbf{I})$ and iteratively applying the learned backward process $p_\theta(x_{t-1}|x_t)$. Specifically, we compute $x_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(x_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}} G_\theta(x_t, t) \right) + \sigma_t z$, where $z \sim \mathcal{N}(0, \mathbf{I})$, for $t = T$ to 1, yielding a new sample x_0 .

In the context of enhancing the generative capabilities of DMs, incorporating a conditional vector provides a strategic augmentation of the model's architecture. By embedding conditional vector, c , within both the embedding and decoder layers of the U-Net structure in the diffusion process, the model gains an additional layer of contextual guidance. This integration is mathematically articulated as $\|z - G_\theta(x_t, t, c)\|^2 = \|z - G_\theta(\sqrt{\alpha_t}x_0 + \sqrt{1 - \bar{\alpha}_t}z, t, c)\|^2$, where the conditional vector c is seamlessly intertwined with the noise prediction and denoising functions of the generative model, G_θ . Such an approach leverages the conditionality to steer the generative process, thereby imbuing the model with enhanced directional specificity and adaptiveness in its generation capabilities, aligning closely with the encoded conditions in c .

Our LDM model operates under a linear beta schedule, which dictates the noise addition and removal process across the diffusion stages. This schedule is precomputed and stored as buffers, allowing for consistent noise manipulation during both training and sampling phases. For this work we use a starting beta value of 1×10^{-4} and a final beta value of 0.02. The diffusion process involves progressively adding noise to the latent features and then denoising them through a series of timesteps to generate the final microstructure.

To guide the diffusion process, the model employs two key embedding networks:

- Time embedding: this network converts the current timestep into an embedding, providing temporal guidance during the denoising phase.
- Context embedding: the context embedding network incorporates manufacturing features that condition the generation process, ensuring that the generated microstructures adhere to specific manufacturing parameters.

During the forward pass, the input 3D data is first encoded through the VAE to extract latent features. These features are then processed by a feature predictor model to obtain context features, specifically the initial four manufacturing features (e.g., two volume fractions and two tortuosities). These latent features are progressively diffused using the predefined beta schedule, with the U-Net model performing denoising at each timestep. The denoising process is informed by both time and context embeddings, enabling precise reconstruction of the microstructure. For new sample generation, the diffusion process is reversed, starting from pure noise and progressively refining the latent space into a structured representation conditioned on the context features.

4.3 Training and inference

As shown in Fig. 10, the training process consists of three steps. First, the VAE is trained on the original training dataset. Once the VAE is trained, we encode the entire training dataset to obtain the latent representation, which becomes the training data for both the feature predictor and the diffusion model. In the second step, we train the feature predictor. Once trained, the input to the feature predictor is a latent representation of the microstructure, and the output is the features of interest, such as manufacturing parameters, tortuosity, volume fraction, etc. Finally, the LDM is trained to denoise and recover the original data from noisy inputs, with the corresponding features of interest used as conditioning. The detail of the training process is provided in the appendix Section A.1.

The inference process begins with the pre-trained weights of the LDM, VAE decoder, and feature predictor. The VAE encoder is not required during inference. The process involves user input and random noise sampled in the latent space. The random noise is iteratively refined by the LDM, conditioned on the user inputs. After 1000 iterations, the denoised latent representation of the microstructure is obtained. This step is the most time-consuming during inference. However, despite this many iterations, the process remains highly efficient because the denoising occurs in latent space rather than pixel



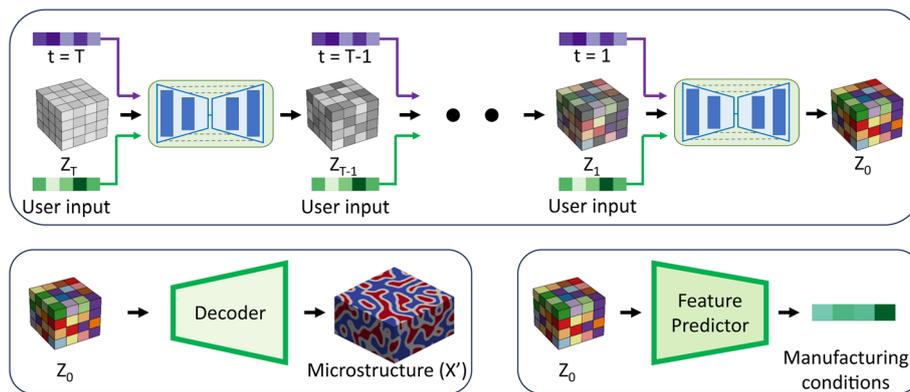


Fig. 11 Overview of the inference framework for the proposed LDM-based model: random noise Z_T is sampled in latent space, and the diffusion model gradually denoises it over T steps. User inputs condition the denoising process. Z_0 is then passed through the VAE decoder and the feature predictor to obtain the microstructure and its manufacturing parameters, respectively.

space, which has 1000 times fewer dimensions. The inference pipeline is demonstrated in Fig. 11. Once the denoised latent representation of the microstructure is obtained, it is passed through both the feature predictor and the VAE decoder. The feature predictor provides the manufacturing conditions, while the VAE decoder generates the final conditioned microstructure. Using NVIDIA A100 80GB GPUs, it takes approximately 2 seconds to generate and save a single microstructure, including export to storage.

Author contributions

Nirmal Baishnab: conceptualization, data curation, formal Analysis, investigation, methodology, software, validation, visualization, writing – original draft, writing – review & editing; Ethan Herron: conceptualization, investigation, methodology, software, validation, writing – original draft, writing – review & editing; Aditya Balu: conceptualization, methodology, validation; Soumik Sarkar: project administration, supervision, validation; Adarsh Krishnamurthy: project administration, supervision, validation, visualization, writing – original draft, writing – review & editing; Baskar Ganapathysubramanian: funding acquisition, methodology, project administration, resources, supervision, validation, visualization, writing – original draft, writing – review & editing.

Conflicts of interest

The authors declare that they have no conflict of interest with respect to the contents of this article.

Data availability

Microgen3D is a dataset of 3D voxelized microstructures designed for training, evaluation, and benchmarking of generative models—especially Conditional Latent Diffusion Models (LDMs). It includes both synthetic (Cahn–Hilliard) and experimental microstructures with multiple phases (2 to 3). The voxel grids range from $64 \times 64 \times 64$ up to $128 \times 128 \times 64$.

The dataset consists of three microstructure types:

- Experimental microstructures ($64 \times 64 \times 64$): voxelized from real-world samples for modeling.
- 2-phase Cahn–Hilliard microstructures ($128 \times 128 \times 64$): thresholded from Cahn–Hilliard simulations.
- 3-phase Cahn–Hilliard microstructures ($128 \times 128 \times 64$): thresholded from Cahn–Hilliard simulations.

For each dataset type, pretrained generative model weights are provided:

- `vae.pt` – variational autoencoder
- `fp.pt` – feature predictor
- `ddpm.pt` – denoising diffusion probabilistic model

In addition to the full datasets, smaller sample subsets are provided for testing and demonstration purposes.

All datasets and pretrained weights have been permanently archived on Zenodo: <https://doi.org/10.5281/zenodo.17010419>. The complete codebase has also been archived on Zenodo: <https://doi.org/10.5281/zenodo.17029570>.

For convenience, the dataset is additionally available on Hugging Face: <https://huggingface.co/datasets/BGLab/microgen3D>, and the latest development version of the code is available on GitHub: <https://github.com/baskargroup/MicroGen3D>.

Supplementary information: the training loss curves, additional inference examples, the feature distribution of the training data, and results from extended conditioning experiments. See DOI: <https://doi.org/10.1039/d5dd00159e>.

A Appendices

A.1 Training process details and hyperparameter tuning

All three components of the architecture—the VAE, feature predictor, and LDM—were trained for 500 epochs with a batch size of 32. The batch size of 32 is large enough to fully utilize GPU parallelism while being small enough to avoid out-of-memory errors, particularly given the use of 3D data. The Adam optimizer⁵⁵ was employed for gradient-based optimization due to its widespread adoption, stability, and efficiency. The learning rate was dynamically adjusted using a cosine annealing scheduler, which gradually reduces the learning rate to effectively minimize the loss.^{56,57} The training time and loss



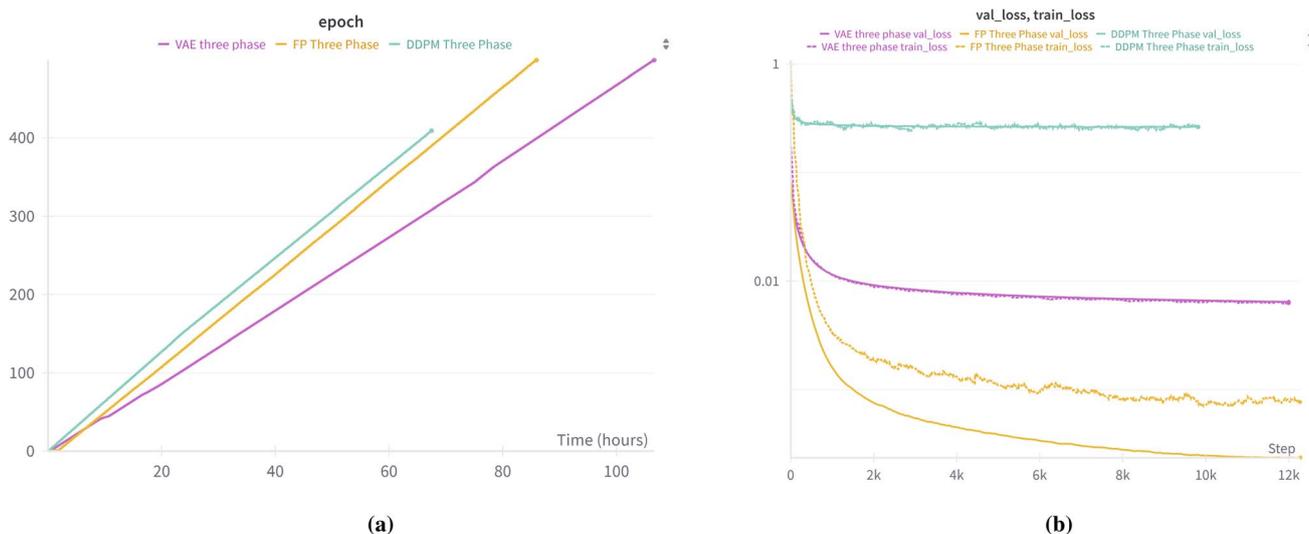


Fig. 12 Training log of the LDM. (a) Epoch progression over wall-clock training time. (b) Training and validation loss curves for all three components of the LDM framework.

curves are shown in Fig. 12 for the CH three-phase dataset. Training the VAE, feature predictor, and DDPM required approximately 100, 80, and 60 hours, respectively. The DDPM was stopped early at 400 epochs, as further training yielded minimal improvements in both training and validation loss. In total, the complete training process required approximately 250 hours, or nearly 11 days. Note that the training loss for the feature predictor appears higher because dropout regularization was applied during training. Fig. 12 is produced using Wandb.⁵⁸

The loss function for VAE combined a Mean Squared Error (MSE) loss for reconstruction and a Kullback–Leibler Divergence (KLD) loss,⁵⁹ with a weight of 1×10^{-6} for regularizing the latent space. The goal was to keep both the KLD and reconstruction losses in the same order of magnitude. The feature predictor was trained using an MSE loss function to assess the accuracy of predictions by measuring the difference between predicted and actual feature values. The encoder of the pre-trained VAE was kept frozen during feature predictor training phase. For both the VAE and the feature predictor, the initial learning rate was set to 5×10^{-5} , with a minimum of 5×10^{-7} .

For the LDM, the diffusion process was divided into 1000 timesteps. The training objective was to minimize the MSE between the predicted noise and the actual noise added during the diffusion process. Initial and minimum learning rates are 1×10^{-6} and 1×10^{-7} , respectively. The learning rate was selected based on the pioneering work by,²⁶ which demonstrated the effectiveness of using this order of magnitude in similar architectures. Both VAE and feature predictor were kept frozen during LDM training.

The training process for all models was conducted in a GPU-enabled environment, using an NVIDIA A100 GPU with 80 GB of memory. The entire framework was implemented in PyTorch and managed by PyTorch Lightning, which handled the training loop, logging, and checkpointing. Checkpoints were automatically saved based on the validation loss, ensuring that only the best-performing models were retained. Throughout the training, real-time progress and performance metrics were continuously logged using the WandB⁵⁸ logger, providing detailed experiment tracking and facilitating reproducibility and scalability.

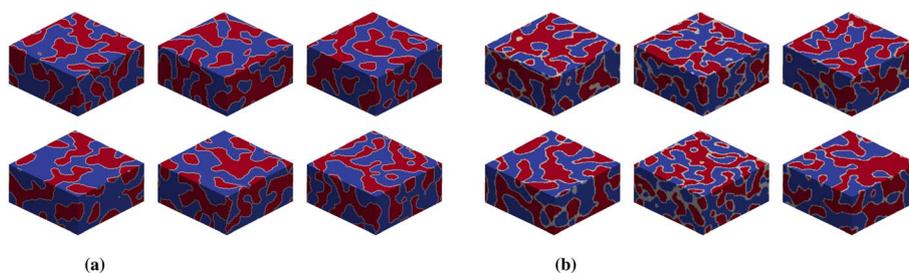


Fig. 13 Representative microstructures generated using only the VAE component of our framework, where random noise was sampled in the latent space and decoded using the VAE decoder. Consistent with the known behavior of VAEs, these outputs exhibit distorted, lower-quality features with inconsistent interfaces and reduced morphological sharpness. Additionally, without the diffusion stage and conditioning, control over target microstructural descriptors is not possible. (a) CH two-phase VAE-only generation. (b) CH three-phase VAE-only generation.



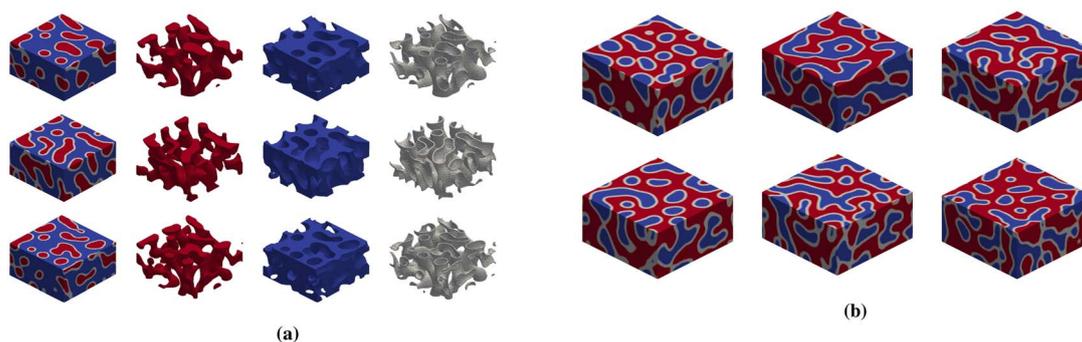


Fig. 14 Representative three-phase microstructures generated by the full conditional LDM framework, exhibiting more consistent interfaces, sharper morphological features, and controlled generation compared to the VAE-only results. (a) Sampled microstructures with a predominant phase B (volume fraction above 0.5). (b) Microstructures generated from the same conditional features: volume fraction of phase A and phase B mix 0.3 and 0.2, respectively. Tortuosity of both phases is 0.3.

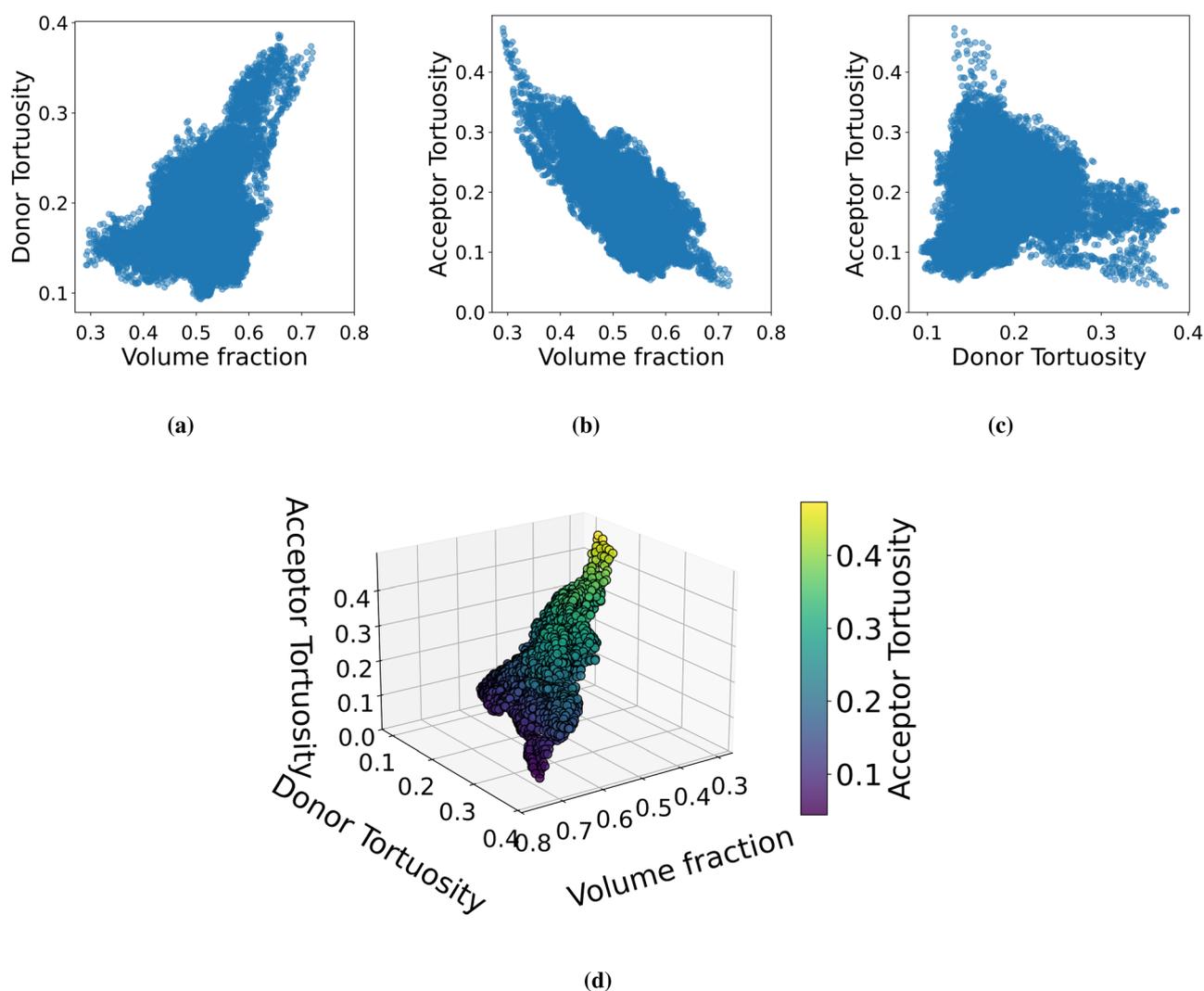


Fig. 15 Distribution of all three features of interest. (a), (b), and (c) show pairwise distributions, while (d) presents a three-dimensional plot of all three features. This visualization highlights the range of the features, and how they are distributed relative to one another. (a) Donor tortuosity vs volume fraction. (b) Acceptor tortuosity vs volume fraction. (c) Acceptor tortuosity vs donor tortuosity. (d) 3D scatter plot of all three features.



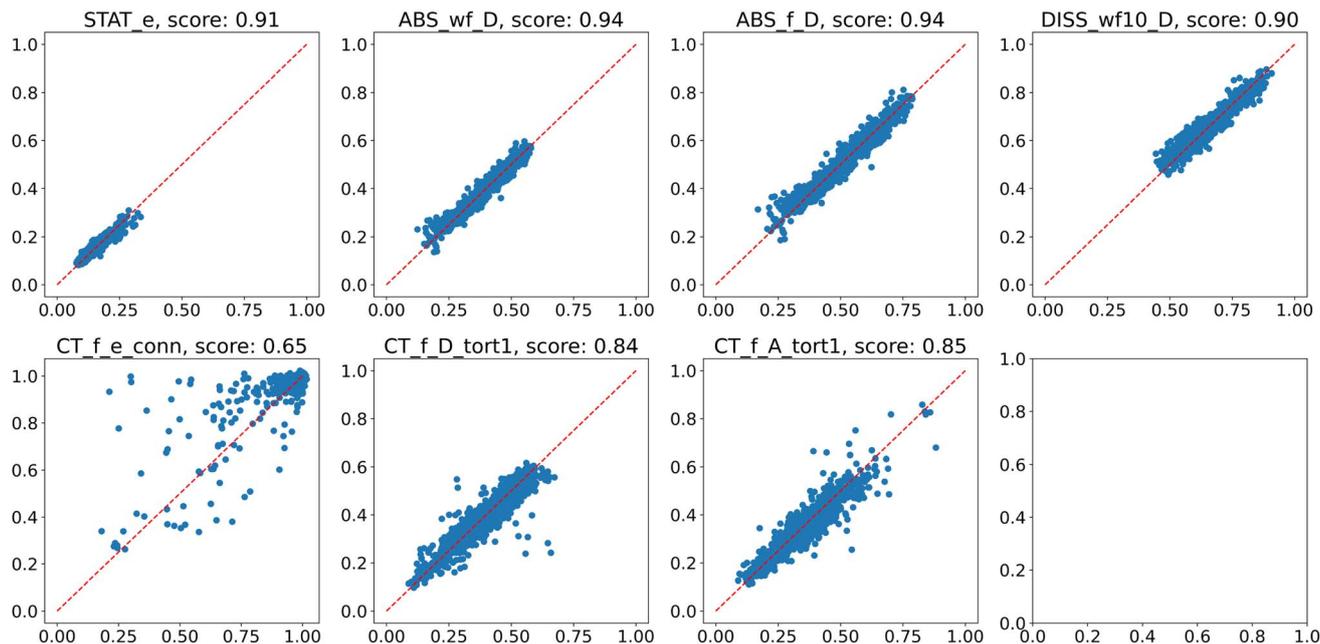


Fig. 16 Scatter plots comparing input and measured features for the seven conditioning variables. The red dashed lines indicate perfect agreement ($y = x$). The R^2 values for each feature are shown within each subplot.

A.2 Inference microstructure samples

Fig. 13 shows some examples of two-phase and three-phase microstructures generated using only the VAE decoder. The results show distorted features and inconsistent interfaces, indicating the limitations of VAE-only generation without diffusion or conditioning.

Fig. 14 shows some additional examples of three-phase microstructures generated by the conditional LDM, demonstrating sharper features and controlled generation, including cases with predominant phase B and specified volume fractions/tortuosities.

A.3 Experimental training dataset feature distribution

Fig. 15 shows the distribution of donor tortuosity, acceptor tortuosity, and volume fraction in the experimental dataset, with pairwise plots and a 3D scatter plot. These three features are the LDM conditioning features.

A.4 Test with seven conditional features

Fig. 16 shows the correlation between the target input features and the measured features for the generated microstructures when conditioning on seven parameters. As the number of conditioning parameters increases, the model maintains reasonably strong correlations. However, a moderate decline in R^2 values is observed due to the increased complexity and sparsity of the conditioning space. To know about these microstructural features, see this here⁶⁰

Acknowledgements

This work was supported by the National Science Foundation under CMMI-2053760 and DMR-2323716. We acknowledge computing support from NSF ACCESS.

References

- 1 R. E. Newnham, *Structure-property relations*, Springer Science & Business Media, vol. 2, 2012.
- 2 T. Le, V. Chandana Epa, F. R. Burden and D. A. Winkler, Quantitative structure–property relationship modeling of diverse materials properties, *Chem. Rev.*, 2012, **112**(5), 2889–2919.
- 3 C. E. Carraher Jr and R. B. Seymour, *Structure—property relationships in polymers*, Springer Science & Business Media, 2012.
- 4 J. Li, Q. Zhang, R. Huang, X. Li and H. Gao, Towards understanding the structure–property relationships of heterogeneous-structured materials, *Scr. Mater.*, 2020, **186**, 304–311.
- 5 A. M. Paul and R. E. Dunin-Borkowski, Electron tomography and holography in materials science, *Nat. Mater.*, 2009, **8**(4), 271–280.
- 6 M. C. Scott, C.-C. Chen, M. Mecklenburg, C. Zhu, R. Xu, P. Ercius, D. Ulrich, B. C. Regan and J. Miao, Electron tomography at 2.4-ångström resolution, *Nature*, 2012, **483**(7390), 444–447.
- 7 L. E. Franken, E. J. Boekema and M. C. A. Stuart, Transmission electron microscopy as a tool for the characterization of soft materials: application and interpretation, *Adv. Sci.*, 2017, **4**(5), 1600476.



- 8 M. Azad and A. Abdullah, Scanning electron microscopy (sem): A review. in. *Proceedings of the 2018 International Conference on Hydraulics and Pneumatics*, HERVEX, Băile Govora, Romania, vol. 2018, 2018, pp. 7–9.
- 9 R. Bostanabad, Y. Zhang, X. Li, T. Kearney, L. C. Brinson, D. W. Apley, W. K. Liu and W. Chen, Computational microstructure characterization and reconstruction: Review of the state-of-the-art techniques, *Prog. Mater. Sci.*, 2018, **95**, 1–41.
- 10 S. Torquato and H. W. Haslach Jr, Random heterogeneous materials: microstructure and macroscopic properties, *Appl. Mech. Rev.*, 2002, **55**(4), B62–B63.
- 11 R. Bostanabad, A. T. Bui, W. Xie, D. W. Apley and W. Chen, Stochastic microstructure characterization and reconstruction *via* supervised learning, *Acta Mater.*, 2016, **103**, 89–102.
- 12 Z. Jiang, W. Chen and C. Burkhart, Efficient 3d porous microstructure reconstruction *via* gaussian random field and hybrid optimization, *J. Microsc.*, 2013, **252**(2), 135–148.
- 13 H. Xu, D. A. Dikin, C. Burkhart and W. Chen, Descriptor-based methodology for statistical characterization and 3d reconstruction of microstructural materials, *Comput. Mater. Sci.*, 2014, **85**, 206–216.
- 14 Y. Jiao, F. H. Stillinger and S. Torquato, Modeling heterogeneous materials *via* two-point correlation functions. ii. algorithmic details and applications, *Phys. Rev. E*, 2008, **77**(3), 031135.
- 15 A. Bandi, P. V. S. R. Adapa and Y. E. V. P. Kumar Kuchi, The power of generative ai: A review of requirements, models, input–output formats, evaluation metrics, and challenges, *Future Internet*, 2023, **15**(8), 260.
- 16 Y. Cao, S. Li, Y. Liu, Z. Yan, Y. Dai, P. S. Yu, and L. Sun, A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt, *arXiv*, 2023, preprint, arXiv:2303.04226, DOI: [10.48550/arXiv.2303.04226](https://doi.org/10.48550/arXiv.2303.04226).
- 17 D. P. Kingma and M. Welling, Auto-encoding variational bayes, *arXiv*, 2013, preprint, arXiv:1312.6114, DOI: [10.48550/arXiv.1312.6114](https://doi.org/10.48550/arXiv.1312.6114).
- 18 I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, Generative adversarial networks, *Commun. ACM*, 2020, **63**(11), 139–144.
- 19 J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan and S. Ganguli, Deep unsupervised learning using nonequilibrium thermodynamics, in. *International conference on machine learning*, PMLR, 2015, pp. 2256–2265.
- 20 Z. Wang, S. Qi and T. E. Ward, Generative adversarial networks in computer vision: A survey and taxonomy, *ACM Comput. Surv.*, 2021, **54**(2), 1–38.
- 21 A. Henkes and H. Wessels, Three-dimensional microstructure generation using generative adversarial neural networks in the context of continuum micromechanics, *Comput. Methods Appl. Mech. Eng.*, 2022, **400**, 115497.
- 22 T. Hsu, W. K. Epting, H. Kim, H. W. Abernathy, G. A. Hackett, A. D. Rollett, P. A. Salvador and E. A. Holm, Microstructure generation *via* generative adversarial network for heterogeneous, topologically complex 3d materials, *JOM*, 2021, **73**, 90–102.
- 23 S. Chun, S. Roy, Y. T. Nguyen, J. B. Choi, H. S. Udaykumar and S. S. Baek, Deep learning for synthetic microstructure generation in a materials-by-design framework for heterogeneous energetic materials, *Sci. Rep.*, 2020, **10**(1), 13307.
- 24 T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford and X. Chen, Improved techniques for training gans, *Adv. Neural Inf. Process. Syst.*, 2016, **29**.
- 25 P. Dhariwal and A. Nichol, Diffusion models beat gans on image synthesis, *Adv. Neural Inf. Process. Syst.*, 2021, **34**, 8780–8794.
- 26 R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, High-resolution image synthesis with latent diffusion models, in. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10684–10695.
- 27 F.-A. Croitoru, V. Hondru, R. T. Ionescu and M. Shah, Diffusion models in vision: A survey, *IEEE Trans. Pattern Anal. Mach. Intell.*, 2023, **45**(9), 10850–10869.
- 28 P. Du, M. H. Parikh, X. Fan, X.-Y. Liu and J.-X. Wang, Conditional neural field latent diffusion model for generating spatiotemporal turbulence, *Nat. Commun.*, 2024, **15**(1), 10416.
- 29 E. Herron, J. Rade, A. Jignasu, B. Ganapathysubramanian, A. Balu, S. Sarkar, and A. Krishnamurthy, Latent diffusion models for structural component design, *arXiv*, 2023, preprint, arXiv:2309.11601, DOI: [10.48550/arXiv.2309.11601](https://doi.org/10.48550/arXiv.2309.11601).
- 30 W. H. L. Pinaya, P.-D. Tudosiu, J. Dafflon, P. F. Da Costa, V. Fernandez, P. Nachev, S. Ourselin and M. J. Cardoso, Brain imaging generation with latent diffusion models, in. *MICCAI Workshop on Deep Generative Models*, Springer, 2022, pp. 117–126.
- 31 A. Blattmann, R. Rombach, H. Ling, T. Dockhorn, S. W. Kim, S. Fidler and K. Kreis, Align your latents: High-resolution video synthesis with latent diffusion models, in. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 22563–22575.
- 32 K.-H. Lee and G. J. Yun, Microstructure reconstruction using diffusion-based generative models, *Mech. Adv. Mater. Struct.*, 2024, **31**(18), 4443–4461.
- 33 X. Lyu and X. Ren, Microstructure reconstruction of 2d/3d random materials *via* diffusion-based deep generative models, *Sci. Rep.*, 2024, **14**(1), 5041.
- 34 P. Fernandez-Zelaia, J. Cheng, J. Mayeur, A. K. Ziabari and M. M. Kirka, Digital polycrystalline microstructure generation using diffusion probabilistic models, *Materialia*, 2024, **33**, 101976.
- 35 E. Herron, X. Y. Lee, A. Balu, B. S. S. Pokuri, B. Ganapathysubramanian, S. Sarkar and A. Krishnamurthy, Generative design of material microstructures for organic solar cells using diffusion models, in. *AI for Accelerated Materials Design NeurIPS 2022 Workshop*, 2022.
- 36 Z. Gao, C. Zhu, C. Wang, Y. Shu, S. Liu, J. Miao and L. Yang, Advanced deep learning framework for multi-scale



- prediction of mechanical properties from microstructural features in polycrystalline materials, *Comput. Methods Appl. Mech. Eng.*, 2025, **438**, 117844.
- 37 K.-H. Lee and G. J. Yun, Denoising diffusion-based synthetic generation of three-dimensional (3d) anisotropic microstructures from two-dimensional (2d) micrographs, *Comput. Methods Appl. Mech. Eng.*, 2024, **423**, 116876.
- 38 C. J. Kuehmann and G. B. Olson, Computational materials design and engineering, *Mater. Sci. Technol.*, 2009, **25**(4), 472–478.
- 39 J. H. Panchal, S. R. Kalidindi and D. L. McDowell, Key computational modeling issues in integrated computational materials engineering, *Comput.-Aided Des.*, 2013, **45**(1), 4–25.
- 40 S. S. Lee and Y.-L. Loo, Structural complexities in the active layers of organic electronics, *Annu. Rev. Chem. Biomol. Eng.*, 2010, **1**(1), 59–78.
- 41 F. Liu, Yu Gu, J. W. Jung, W. H. Jo and T. P. Russell, On the morphology of polymer-based photovoltaics, *J. Polym. Sci., Part B: Polym. Phys.*, 2012, **50**(15), 1018–1044.
- 42 M. C. Heiber, K. Kister, A. Baumann, V. Dyakonov, C. Deibel and T.-Q. Nguyen, Impact of tortuosity on charge-carrier transport in organic bulk heterojunction blends, *Phys. Rev. Appl.*, 2017, **8**(5), 054043.
- 43 O. Wodo, J. D. Roehling, A. J. Moulé and B. Ganapathysubramanian, Quantifying organic solar cell morphology: a computational study of three-dimensional maps, *Energy Environ. Sci.*, 2013, **6**(10), 3060–3070.
- 44 O. Wodo and B. Ganapathysubramanian, Computationally efficient solution to the cahn–hilliard equation: Adaptive implicit time schemes, mesh sensitivity analysis and the 3d isoperimetric problem, *J. Comput. Phys.*, 2011, **230**(15), 6037–6060.
- 45 A. Vondrous, M. Selzer, J. Hötzer and B. Nestler, Parallel computing for phase-field models, *Int. J. High Perform. Comput. Appl.*, 2014, **28**(1), 61–72.
- 46 Y. Li, Y. Choi and J. Kim, Computationally efficient adaptive time step method for the cahn–hilliard equation, *Comput. Math. Appl.*, 2017, **73**(8), 1855–1864.
- 47 M. C. Heiber, A. A. Herzing, L. J. Richter and D. M. DeLongchamp, Charge transport and mobility relaxation in organic bulk heterojunction morphologies derived from electron tomography measurements, *J. Mater. Chem. C*, 2020, **8**(43), 15339–15350.
- 48 A. A. Herzing, L. J. Richter and I. M. Anderson, 3d nanoscale characterization of thin-film organic photovoltaic device structures *via* spectroscopic contrast in the tem, *J. Mater. Chem. C*, 2010, **114**(41), 17501–17508.
- 49 O. Wodo, S. Tirthapura, S. Chaudhary and B. Ganapathysubramanian, A graph-based formulation for computational characterization of bulk heterojunction morphology, *Org. Electron.*, 2012, **13**(6), 1105–1113.
- 50 Owodo Lab, *Graspi*, <https://owodolab.github.io/graspi/listOfDescriptors.html>, accessed: 2025-06-09.
- 51 O. J. J. Ronsin and J. Harting, Formation of crystalline bulk heterojunctions in organic solar cells: insights from phase-field simulations, *ACS Appl. Mater. Interfaces*, 2022, **14**(44), 49785–49800.
- 52 B. König, O. J. J. Ronsin and J. Harting, Two-dimensional cahn–hilliard simulations for coarsening kinetics of spinodal decomposition in binary mixtures, *Phys. Chem. Chem. Phys.*, 2021, **23**(43), 24823–24833.
- 53 A. Bentamou, S. Chrétien and Y. Gavet, 3d denoising diffusion probabilistic models for 3d microstructure image generation of fuel cell electrodes, *Comput. Mater. Sci.*, 2025, **248**, 113596.
- 54 J. Ho, A. Jain and P. Abbeel, Denoising diffusion probabilistic models, in: *Advances in Neural Information Processing Systems*, ed. H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan and H. Lin, vol. 33, 2020, pp. 6840–6851.
- 55 P. K. Diederik, Adam: A method for stochastic optimization, *arXiv*, 2014, preprint, arXiv:1412.6980, DOI: [10.48550/arXiv.1412.6980](https://doi.org/10.48550/arXiv.1412.6980).
- 56 I. Loshchilov and F. Hutter, Sgdr: Stochastic gradient descent with warm restarts, *arXiv*, 2016, preprint, arXiv:1608.03983, DOI: [10.48550/arXiv.1608.03983](https://doi.org/10.48550/arXiv.1608.03983).
- 57 I. Loshchilov, Decoupled weight decay regularization, *arXiv*, 2017, preprint arXiv:1711.05101, DOI: [10.48550/arXiv.1711.05101](https://doi.org/10.48550/arXiv.1711.05101).
- 58 L. Biewald, *Experiment tracking with weights and biases*, 2020, <https://wandb.ai/site>, accessed: 2025-06-15.
- 59 K. Solomon and R. A. Leibler, On information and sufficiency, *Ann. Math. Stat.*, 1951, **22**(1), 79–86.
- 60 Owodo Lab, *Graspi list of descriptors*, 2020, <https://owodolab.github.io/graspi/listOfDescriptors.html>, accessed: 2025-01-20.

