3D Multiphase Heterogeneous Microstructure Generation Using Conditional Latent Diffusion Models

Nirmal Baishnab[1], **Ethan Herron**[1], **Aditya Balu**[1], **Soumik Sarkar**[1],
Adarsh Krishnamurthy[1], **Baskar Ganapathysubramanian**[1]
[1]*Iowa State University, Ames, IA 50011, USA*
Correspondence: `baskarg@iastate.edu`

Abstract

The development of a microstructure generation framework tailored to user-specific needs is crucial for understanding materials behavior through distinct processing-structure-property relationships. Recent advancements in generative modeling, particularly with Latent Diffusion Models (LDM), have significantly enhanced our ability to create high-quality images that fulfill specific user requirements. In this paper, we present a scalable framework that employs LDM to generate 3D microstructures (128x128x64) with over a million voxels, customized to user-defined statistical and topological features. This framework can also predict manufacturing conditions that produce these microstructures experimentally, solving the reachability issue. Our work focuses on organic photovoltaics (OPV), but the architecture allows for potential extensions into other fields of materials science by adjusting the training dataset.

1 Introduction

In materials science, the mapping between structure and property is a foundational concept, with microstructures often serving as the primary influencers of a material's physical characteristics and performance [1–4]. However, direct experimental observation and reconstruction of these microstructures are frequently hindered by high costs and technical challenges, posing difficulties in accurately evaluating and forecasting material properties [5–8]. A prevalent objective in this field is to generate statistically representative sets of microstructural realizations to aid computational design and performance analysis. Addressing this challenge, we present a 3D microstructure generation framework using state of the art generative artifical intelligence (AI) tools. This framework enables the customized creation of microstructures based on specific requirements, facilitating on-demand microstructure generation to support various research and development activities.

Several statistical approaches have been developed for microstructure generation, such as Markov Random Fields (MRFs) [9], Gaussian Random Fields (GRFs) [10], and descriptors based microstructure reconstruction [11, 12]. Bostanabad et al. [13] offer a detailed overview of current techniques for microstructure reconstruction. While these methods have proven useful, they come with certain limitations. Torquato's book [14] provides foundation insights into characterization of random heterogeneous materials and provides comprehensive overview of mathematical and computational tools for microstructure modeling and generation. Statistical models are computationally intensive and hence not scalable with respect to both the size and the number of microstructure generations. These models are less flexible and often require specific assumptions about the statistical properties of the microstructure, such as stationarity and isotropy. Consequently, they may not generalize effectively across different types of materials or structures. Adapting these models to incorporate new constraints or objectives can be challenging and may requires substantial changes to the model or the optimization process.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

In recent years, advanced generative AI models have demonstrated their powerful capabilities in capturing intricate features from training data distributions. These AI tools are versatile and have applications in various fields [15, 16]. Prominent examples of such generative AI tools include Variational Autoencoders (VAEs) [17], Generative Adversarial Networks (GANs) [18], and Diffusion Models (DMs) [19]. While VAEs are known for their ability to learn efficient data encodings, they often struggle to generate sharp images [20]. GANs have been successfully employed in generating good quality 3D microstructures [21–23]; however, they do not allow users to control the generated microstructures. Furthermore, GANs are known for their training instability [24] and are computationally intensive, especially when generating 3D structures. DMs have been claimed to produce higher-quality images than GANs, showcasing a potential advantage in terms of output quality [25]. However, similar to GANs, DMs also require significant computational resources, particularly during inference, due to their iterative nature of the generation process [25].

Considering the limitations of VAEs, DMs, and GANs, the LDMs have emerged as a state-of-the-art solution, combining the strengths of VAEs and DMs [26, 27]. LDMs operate within a latent space, a reduced dimensionality achieved by a VAE within the LDM framework, which is significantly less computationally intensive than operating in pixel space. This architecture enables LDMs to generate high-quality, diverse, and detailed samples, offering a computational efficiency that is several orders of magnitude greater than that of traditional DMs [26]. LDMs enable a controlled generation process by specifying attributes or conditions, crucial for creating customized microstructure designs [26, 27]. Unlike GANs, which may encounter training challenges and mode collapse, leading to a constrained output variety, LDMs demonstrate enhanced stability during training and a more varied output. This dependability and adaptability make LDMs a suitable choice across a broad spectrum of applications in the field of generative AI [28–31]. Previously, 2D microstructures of organic solar cells (OSC) were designed using LDM [32] and this approach provided no control over the sampling process. In our work, we address the major challenges associated with 3D microstructure generation by leveraging LDM. We introduce an efficient LDM-based framework for 3D microstructure generation that not only facilitates the on-demand creation of microstructures tailored to user specifications but also predicts relevant manufacturing parameters for the generated microstructures. In this study, we limit the user specifications to volume fraction and tortuosity. In the context of OPVs, both the donor volume fraction of the active layer and tortuosity are crucial[33, 34]. The volume fraction refers to the ratio of donor to acceptor materials within the active layer, impacting the efficiency of charge generation and transport. Tortuosity, on the other hand, describes the complexity of the pathways that charges (electrons and holes) must navigate through within the active layer to reach the electrodes. To quantify these parameters, we employ the method described in Wodo et al. [35].

Our specific contributions reflect these major challenges and are as follows:

- **3D diverse microstructure generation:** We employ a latent diffusion model to generate multiphase 3D microstructures with more than a *million voxels*, and illustrate for both two-phase and three-phase designs.
- **Enable conditional microstructure generation:** We can generate microstructures with *desired volume fractions and tortuosities*, ensuring alignment with user specifications.
- **Provide manufacturing parameters:** Our framework is also capable of providing manufacturing parameters for the generated microstructures. This will facilitate the transition from conceptualization to actual manufacturing.

2 Methodology

2.1 Training dataset

The 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, carefully captured through simulations under varying conditions defined by two critical parameters: the initial volume fraction (ϕ) and the interaction parameter (χ).

The Cahn-Hilliad 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 insights into how initial concentration heterogeneities influence the dynamics of phase separation. The interaction parameter, χ, is another key variable in the dataset. It

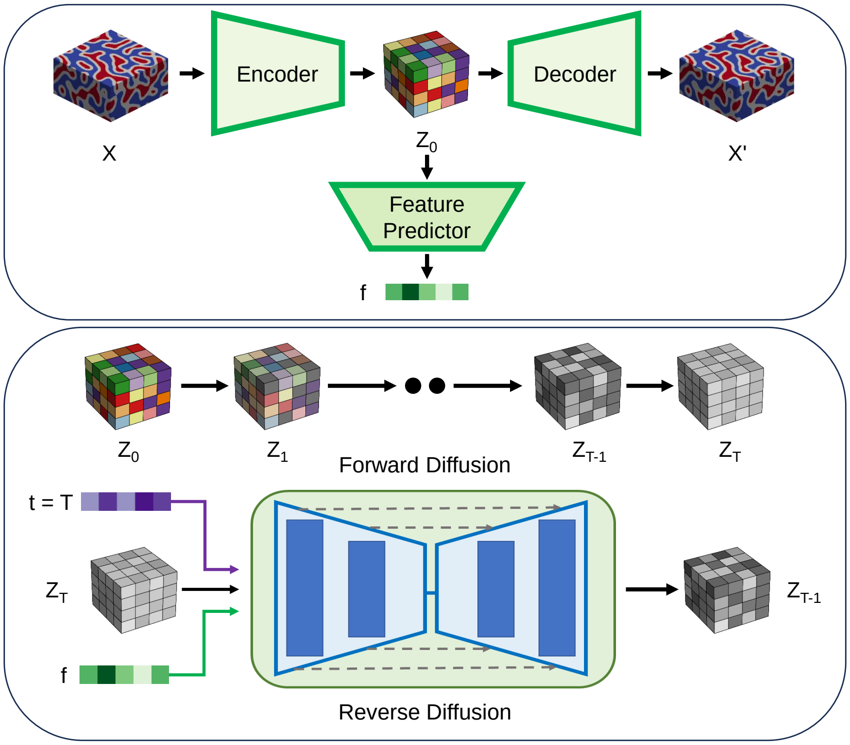


Figure 1: Overview of the proposed LDM-based framework's three-step training process: VAE training and latent representation dataset creation, training of the FP, training of DM in the latent space

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 unfavorable interactions, while a lower value suggests better miscibility. By altering χ, we probe different interaction regimes, from weak to strong phase-separating tendencies. This variation allows for an in-depth examination of how molecular interactions impact the macroscopic patterns formed during phase separation.

For each combination of ϕ and χ, the dataset captures over 400 time-stamped snapshots of a 3D Cahn-Hilliard simulation at 128×128 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. The detailed dataset preparation is provided in the supplementary section.

2.2 Generative model architecture

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. The architecture of the training framework is provided in Figure 1.

2.2.1 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 $\mathbf{z}$ from the latent space, the VAE uses a random sample ϵ drawn from a standard normal distribution:

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

where $\odot$ denotes element-wise multiplication.

The encoder maps the input $\mathbf{x}$ to two parameters in the latent space - the mean μ and the log-variance (log-var):

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

The decoder maps the latent representation $\mathbf{z}$ back to the input space:

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

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

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

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 employ its encoder to compress microstructures with over a million voxels into a more compact encoded representation, sized at 1024 (4x8x8x4). This reduced-dimensional latent space, distinguished by its efficiently learned data distribution, facilitates more efficient and stable diffusion processes.

2.2.2 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.

2.2.3 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. This process is described by the Markov chain:

$$q(x_t \mid x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}x_{t-1}, \beta_t\mathbf{I}) \tag{5}$$

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 $q(x_{t-1} \mid x_t)$. Direct sampling from $q(x_{t-1} \mid x_t)$ is not possible because that would require the complete knowledge of the distribution. Therefore, the model uses a neural network $G_\theta(x_{t-1} \mid x_t)$, 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:

$$\|z - G_\theta(x_t, t)\|^2 = \|z - G_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}z, t)\|^2 \tag{6}$$

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.

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 $\|z - G_\theta(x_t, t, c)\|^2 = \|z - G_\theta(\sqrt{\bar{\alpha}_t}x_0 + \sqrt{1-\bar{\alpha}_t}z, t, c)\|^2$, 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. 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.

2.3 Training and inference

As shown in figure Figure 1, 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

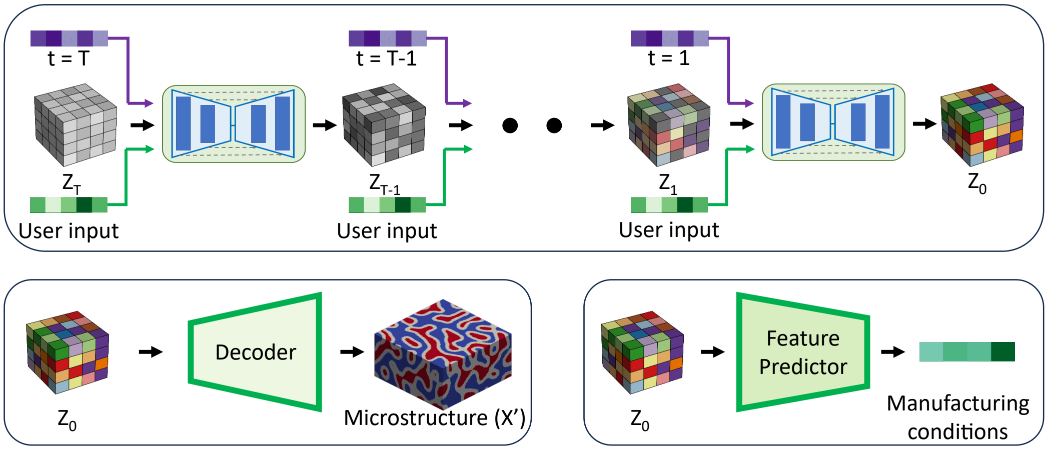


Figure 2: Overview of the inference framework for the proposed LDM-based model: Random noise Z_T is sampled in latent space, and the diffusion model gradually denoises it over T steps. User inputs condition the denoising process. Z_T is then passed through the VAE decoder and the feature predictor to obtain the microstructure and its manufacturing parameters, respectively.

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.

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 1,000 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 space, which has 1,000 times fewer dimensions. The inference pipeline is demonstrated in the Figure 2. 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 GPU cards, it takes 2 seconds to infer a single microstructure.

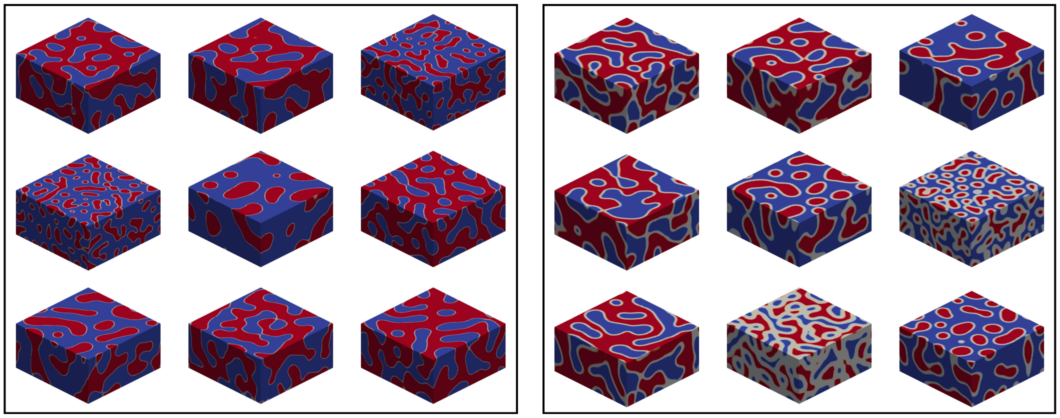
3 Results and Discussion

3.1 Sampling quality

Figure 3 displays samples of microstructures generated by our LDMs, which were trained separately on two-phase and three-phase systems. In the two-phase system, blue represents polymer type A, and red represents polymer type B, corresponding to the donor and acceptor in OPVs, respectively. For the three-phase system, gray denotes the interface between the donor and acceptor. Each microstructure spans a volume of $128 \times 128 \times 64$ voxels, showcasing a broad range of complex features. The trained models generate numerous unique microstructures resembling those in the training set. Traditional generation using numerical solvers of 3D microstructures of this size can require hours or days on substantial amount of computing resources [36–38]. In contrast, our LDM framework produces each microstructure in 0.5 seconds on an NVIDIA A100 GPU, including data export time, highlighting its computational efficiency and potential for GenAI applications.

The transition from two-phase to three-phase systems maintains the quality of the features, suggesting that the framework could be extended to accommodate more than three phases. This result demonstrates the LDM's capability to generate a diverse array of multiphase 3D microstructures, addressing our first contribution. Our model efficiently produces microstructures with over a million voxels ($128 \times 128 \times 64 = 1,048,576$ voxels), offering sufficiently high resolution to capture complex two-phase and three-phase structures. For comparison, [39] designed structures with approximately $80 \times 80 \times 80$ voxels, while [22] reported microstructure sizes of $96 \times 96 \times 96$ voxels.

Our model is designed to be scalable and generalizabley. By simply retraining with a three-phase dataset—without altering the framework—it successfully generated high-quality microstructures featuring smaller domains and more intricate features in three-phase systems. This adaptability suggests that the framework could potentially be extended to accommodate even more phases, depending on the application.



(a) Two phase

(b) Three phase

Figure 3: Samples from LDMs trained on (a) two phase and (b) three phase microstructures.

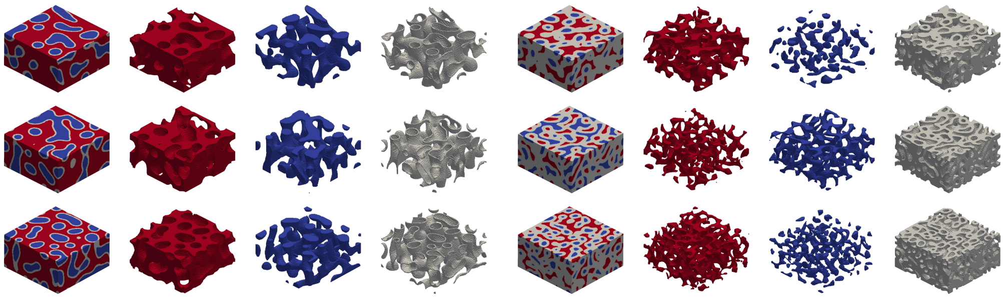
3.2 Conditional sampling

Conditioning in this context refers to a method where additional information, known as a conditioning vector, is provided to the model to guide the generation of microstructures towards desired characteristics. We implemented conditioning by integrating the conditioning vector into the embedding layers of the U-Net[40] backbone within the LDMs. Details on the conditioning process are provided in the supplementary section. Users can specify a list of desired parameter values for conditioning. The LDMs presented here can accommodate volume fractions and tortuosities of both phase A and phase mix as conditioning parameters. The framework can be extended to include additional features, depending on the application (see supplementary results). Figure 4 showcases examples of conditional microstructure generation, demonstrating how the system can create microstructures customized to specific volume fractions and tortuosities based on user inputs. Given the challenge of interpreting the internal structures of 3D microstructures, the images display both non-thresholded versions for the total microstructure and thresholded versions for the phase A only, phase B only, and mixed components.

To evaluate the performance of conditional sampling, we generated 3200 microstructures based on various user-specified volume fractions and tortuosities. We then compared the features of these generated microstructures with the corresponding user inputs, as illustrated in Figure 5. The figure highlights the LDM's effectiveness, showing a Pearson correlation coefficient (or R^2) of 0.93 or higher, underscoring our method's precision in guiding microstructure generation. The high degree of alignment between the desired and generated microstructures' attributes, such as volume fractions and tortuosities, demonstrates the model's adherence to precise user-defined specifications. This capability is critical for targeted material design and optimization, extending our framework beyond other microstructure generation methods [22, 23].

3.3 Prediction of manufacturing parameters

We sampled 3200 distinct microstructures using the same user inputs: volume fractions of phase A and phase mixed were set at 0.3 and 0.2, respectively, with tortuosity values for phases A and B both specified as 0.3. Samples generated from this identical input, available in the Supporting Information, demonstrate the LDM's capability to produce diverse microstructures from consistent

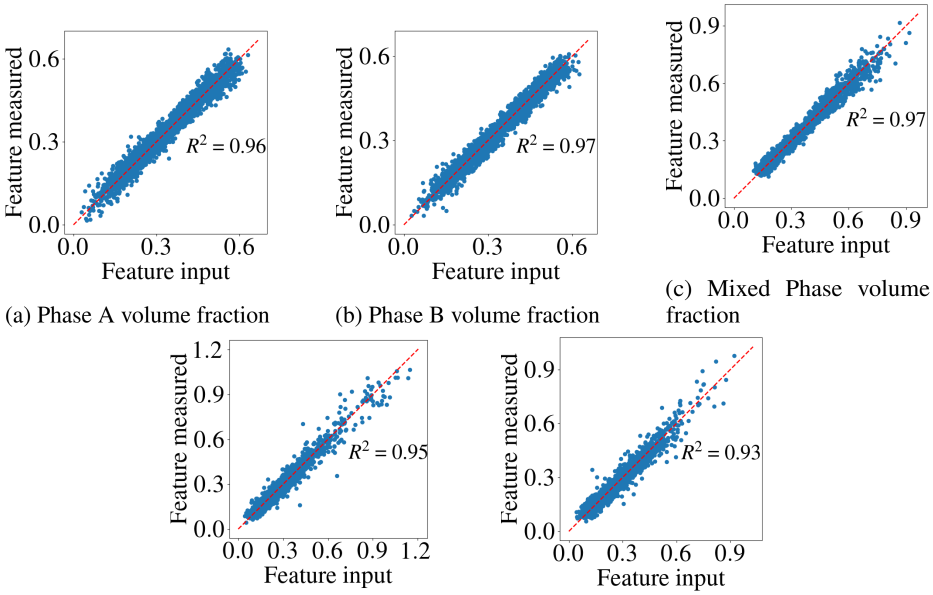


(a) Predominant phase A - more than 0.5

(b) Predominant phase mixed - more than 0.5

Figure 4: 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.

inputs. Figure 6a illustrates the distribution of features extracted from these generated microstructures, where the vertical dotted black lines mark the user inputs. The close alignment between the feature distribution and input values highlights the LDM's precision and conditional consistency. Moreover, Figure 6b presents a contour plot of the manufacturing parameters—χ and timesteps—required to achieve the desired microstructures. This plot identifies two viable χ parameters for the given input, suggesting that both a higher $\chi$with fewer timesteps and a lower χ with more timesteps can produce the targeted microstructures. Our approach predicts the manufacturing parameters (blend ratio, interaction parameter, and annealing time) used by the physics simulator with good accuracy, showcasing a critical integration of computational design with manufacturability. By incorporating actual manufacturing parameters like polymer types, temperature, solvents, annealing time, and pressure into the training dataset, the model could extend its applicability to predict manufacturing conditions for other applications, such as additive manufacturing [41, 42].



(a) Phase A volume fraction

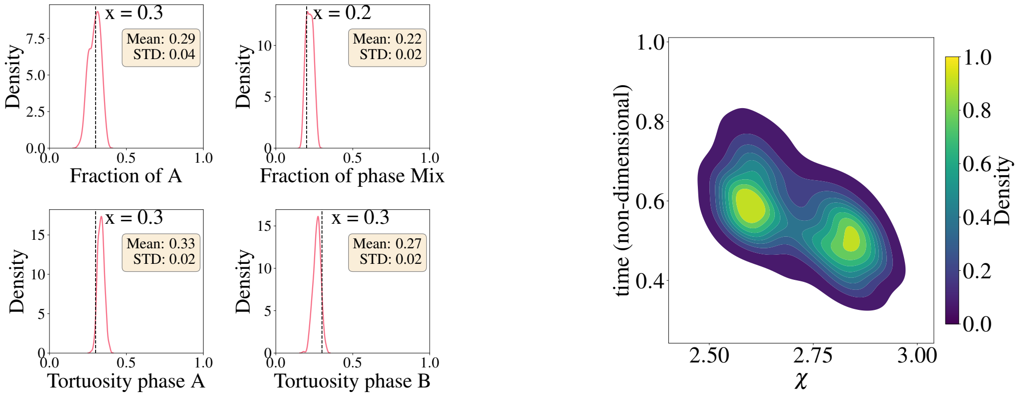
(b) Phase B volume fraction

(c) Mixed Phase volume fraction

(d) Tortuosity A

(e) Tortuosity B

Figure 5: Statistical analysis of conditional microstructure generation: Correlations between all features of interest, user inputs, and the corresponding features measured from generated 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 $\chi$and timesteps for desired microstructure generation.

Figure 6: Variety of microstructures generated by the LDM given identical user inputs. The model can also suggests the manufacturing conditions required to generate such microstructures.

4 Conclusions

Conditional microstructure generation can be useful across various fields, including materials science, energy storage, biomedical engineering, and additive manufacturing, by enabling precise control over microstructure attributes to optimize performance, durability, and functionality. In this work, we presented a versatile, scalable framework for generating 3D microstructures using LDM. Our approach efficiently produces diverse, high-resolution microstructures, such as those in organic photovoltaics, and provides predictive manufacturing parameters to link computational models with experimental synthesis. This capability has the potential to empowers researchers and engineers to design materials with specific properties, advancing our understanding of processing-structure-property relationships.

However, there are a few limitations to consider. The training dataset is generated using physics simulators rather than experimental microstructures. Firstly, our framework shows promising performance on synthetic microstructures but may not achieve the same results on realistic microstructures. The primary reason we did not validate our model with experimental data is the lack of 3D reconstructed microstructure datasets. OPV's complex nanoscale morphologies with interpenetrating polymer and fullerene domains are challenging to capture in 3D. For instance, studies that presented [43, 44] 3D microstructures using tomographic reconstruction techniques have been explored; however, to our knowledge, no published 3D microstructure dataset currently exists. Should 3D structures become available in the future, even in limited quantities, they could serve as valuable resources for fine-tuning our framework. Secondly, the sequential training process—first training the VAE, followed by the feature predictor, and finally the latent diffusion model—could be more streamlined for efficiency and optimization. Thirdly, during inference, users must exercise caution when selecting conditional parameters. Unrealistic user specifications will produce microstructures that do not satisfy the user specifications, which is easily checked using the feature predictor network. To address these limitations, future work could focus on incorporating experimental microstructure data. Optimizing the training pipeline to allow for parallel or integrated training of the VAE, feature predictor, and LDM could significantly reduce the overall training time. Furthermore, developing a more intuitive interface, implementing restrictions among user inputs to prevent unrealistic conditioning parameters, or automating parameter selection could mitigate the risk of generating impractical microstructures, thus making the framework robust and user-friendly for a broader audience.

References

[1] Robert E Newnham. *Structure-property relations*, volume 2. Springer Science & Business Media, 2012.

[2] Tu Le, V Chandana Epa, Frank R Burden, and David A Winkler. Quantitative structure–property relationship modeling of diverse materials properties. *Chemical reviews*, 112(5):2889–2919, 2012.

[3] Charles E Carraher Jr and RB Seymour. *Structure—property relationships in polymers*. Springer Science & Business Media, 2012.

[4] Jianguo Li, Qian Zhang, Ruirui Huang, Xiaoyan Li, and Huajian Gao. Towards understanding the structure–property relationships of heterogeneous-structured materials. *Scripta Materialia*, 186:304–311, 2020.

[5] Paul A Midgley and Rafal E Dunin-Borkowski. Electron tomography and holography in materials science. *Nature materials*, 8(4):271–280, 2009.

[6] MC Scott, Chien-Chun Chen, Matthew Mecklenburg, Chun Zhu, Rui Xu, Peter Ercius, Ulrich Dahmen, BC Regan, and Jianwei Miao. Electron tomography at 2.4-ångström resolution. *Nature*, 483(7390):444–447, 2012.

[7] Linda E Franken, Egbert J Boekema, and Marc CA Stuart. Transmission electron microscopy as a tool for the characterization of soft materials: application and interpretation. *Advanced Science*, 4(5):1600476, 2017.

[8] Azad Mohammed and Avin Abdullah. Scanning electron microscopy (sem): A review. In *Proceedings of the 2018 International Conference on Hydraulics and Pneumatics—HERVEX, Băile Govora, Romania*, volume 2018, pages 7–9, 2018.

[9] Ramin Bostanabad, Anh Tuan Bui, Wei Xie, Daniel W Apley, and Wei Chen. Stochastic microstructure characterization and reconstruction via supervised learning. *Acta Materialia*, 103:89–102, 2016.

[10] Z Jiang, Wei Chen, and C Burkhart. Efficient 3d porous microstructure reconstruction via gaussian random field and hybrid optimization. *Journal of microscopy*, 252(2):135–148, 2013.

[11] Hongyi Xu, Dmitriy A Dikin, Craig Burkhart, and Wei Chen. Descriptor-based methodology for statistical characterization and 3d reconstruction of microstructural materials. *Computational Materials Science*, 85:206–216, 2014.

[12] Yang Jiao, FH Stillinger, and S Torquato. Modeling heterogeneous materials via two-point correlation functions. ii. algorithmic details and applications. *Physical Review E*, 77(3):031135, 2008.

[13] Ramin Bostanabad, Yichi Zhang, Xiaolin Li, Tucker Kearney, L Catherine Brinson, Daniel W Apley, Wing Kam Liu, and Wei Chen. Computational microstructure characterization and reconstruction: Review of the state-of-the-art techniques. *Progress in Materials Science*, 95: 1–41, 2018.

[14] Salvatore Torquato and Henry W Haslach Jr. Random heterogeneous materials: microstructure and macroscopic properties. *Appl. Mech. Rev.*, 55(4):B62–B63, 2002.

[15] Ajay Bandi, Pydi Venkata Satya Ramesh Adapa, and Yudu Eswar Vinay Pratap Kumar Kuchi. The power of generative ai: A review of requirements, models, input–output formats, evaluation metrics, and challenges. *Future Internet*, 15(8):260, 2023.

[16] Yihan Cao, Siyu Li, Yixin Liu, Zhiling Yan, Yutong Dai, Philip S Yu, and Lichao Sun. A comprehensive survey of ai-generated content (aigc): A history of generative ai from gan to chatgpt. *arXiv preprint arXiv:2303.04226*, 2023.

[17] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[19] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pages 2256–2265. PMLR, 2015.

[20] Zhengwei Wang, Qi She, and Tomas E Ward. Generative adversarial networks in computer vision: A survey and taxonomy. *ACM Computing Surveys (CSUR)*, 54(2):1–38, 2021.

[21] Alexander Henkes and Henning Wessels. Three-dimensional microstructure generation using generative adversarial neural networks in the context of continuum micromechanics. *Computer Methods in Applied Mechanics and Engineering*, 400:115497, 2022.

[22] Tim Hsu, William K Epting, Hokon Kim, Harry W Abernathy, Gregory A Hackett, Anthony D Rollett, Paul A Salvador, and Elizabeth A Holm. Microstructure generation via generative adversarial network for heterogeneous, topologically complex 3d materials. *Jom*, 73:90–102, 2021.

[23] Sehyun Chun, Sidhartha Roy, Yen Thi Nguyen, Joseph B Choi, Holavanahalli S Udaykumar, and Stephen S Baek. Deep learning for synthetic microstructure generation in a materials-by-design framework for heterogeneous energetic materials. *Scientific reports*, 10(1):13307, 2020.

[24] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.

[25] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

[26] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[27] Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[28] Pan Du, Meet Hemant Parikh, Xiantao Fan, Xin-Yang Liu, and Jian-Xun Wang. Confild: Conditional neural field latent diffusion model generating spatiotemporal turbulence, 2024.

[29] Ethan Herron, Jaydeep Rade, Anushrut Jignasu, Baskar Ganapathysubramanian, Aditya Balu, Soumik Sarkar, and Adarsh Krishnamurthy. Latent diffusion models for structural component design. *arXiv preprint arXiv:2309.11601*, 2023.

[30] Walter HL Pinaya, Petru-Daniel Tudosiu, Jessica Dafflon, Pedro F Da Costa, Virginia Fernandez, Parashkev Nachev, Sebastien Ourselin, and M Jorge Cardoso. Brain imaging generation with latent diffusion models. In *MICCAI Workshop on Deep Generative Models*, pages 117–126. Springer, 2022.

[31] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten 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*, pages 22563–22575, 2023.

[32] 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.

[33] Feng Liu, Yu Gu, Jae Woong Jung, Won Ho Jo, and Thomas P Russell. On the morphology of polymer-based photovoltaics. *Journal of Polymer Science Part B: Polymer Physics*, 50(15): 1018–1044, 2012.

[34] Michael C Heiber, Klaus Kister, Andreas Baumann, Vladimir Dyakonov, Carsten Deibel, and Thuc-Quyen Nguyen. Impact of tortuosity on charge-carrier transport in organic bulk heterojunction blends. *Physical Review Applied*, 8(5):054043, 2017.

[35] Olga Wodo, John D Roehling, Adam J Moulé, and Baskar Ganapathysubramanian. Quantifying organic solar cell morphology: a computational study of three-dimensional maps. *Energy & Environmental Science*, 6(10):3060–3070, 2013.

[36] Olga Wodo and Baskar Ganapathysubramanian. Computationally efficient solution to the cahn–hilliard equation: Adaptive implicit time schemes, mesh sensitivity analysis and the 3d isoperimetric problem. *Journal of Computational Physics*, 230(15):6037–6060, 2011.

[37] Alexander Vondrous, Michael Selzer, Johannes Hötzer, and Britta Nestler. Parallel computing for phase-field models. *The International journal of high performance computing applications*, 28(1):61–72, 2014.

[38] Yibao Li, Yongho Choi, and Junseok Kim. Computationally efficient adaptive time step method for the cahn–hilliard equation. *Computers & Mathematics with Applications*, 73(8):1855–1864, 2017.

[39] Umar Farooq Ghumman, Akshay Iyer, Rabindra Dulal, Joydeep Munshi, Aaron Wang, TeYu Chien, Ganesh Balasubramanian, and Wei Chen. A spectral density function approach for active layer design of organic photovoltaic cells. *Journal of Mechanical Design*, 140(11):111408, 2018.

[40] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[41] Omar A Abdulrazzaq, Viney Saini, Shawn Bourdo, Enkeleda Dervishi, and Alexandru S Biris. Organic solar cells: a review of materials, limitations, and possibilities for improvement. *Particulate science and technology*, 31(5):427–442, 2013.

[42] Rongming Xue, Jingwen Zhang, Yaowen Li, and Yongfang Li. Organic solar cell materials toward commercialization. *Small*, 14(41):1801793, 2018.

[43] Andrew A Herzing, Lee J Richter, and Ian M Anderson. 3d nanoscale characterization of thin-film organic photovoltaic device structures via spectroscopic contrast in the tem. *The Journal of Physical Chemistry C*, 114(41):17501–17508, 2010.

[44] Michael C Heiber, Andrew A Herzing, Lee J Richter, and Dean M DeLongchamp. Charge transport and mobility relaxation in organic bulk heterojunction morphologies derived from electron tomography measurements. *Journal of Materials Chemistry C*, 8(43):15339–15350, 2020.

[45] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[46] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[47] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[48] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[49] Olivier JJ Ronsin and Jens Harting. Formation of crystalline bulk heterojunctions in organic solar cells: insights from phase-field simulations. *ACS applied materials & interfaces*, 14(44): 49785–49800, 2022.

[50] Björn König, Olivier JJ Ronsin, and Jens Harting. Two-dimensional cahn–hilliard simulations for coarsening kinetics of spinodal decomposition in binary mixtures. *Physical Chemistry Chemical Physics*, 23(43):24823–24833, 2021.

A Appendix

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. We chose a smaller batch size to mitigate the risk of out-of-memory errors, particularly given that we are working with 3D data. The Adam optimizer [45] was employed for gradient-based optimization. The Adam optimizer was selected due to its widespread adoption, stability, and efficiency. The learning rate was dynamically adjusted using a cosine annealing scheduler, which effectively reduces the loss by gradually decreasing the learning rate [46, 47]. Each model took 3-4 days to train, and training all three models sequentially took a total of 11 days.

The loss function for VAE combined a Mean Squared Error (MSE) loss for reconstruction and a Kullback-Leibler Divergence (KLD) loss [48], 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 pretrained 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 WandB logger, providing detailed experiment tracking and facilitating reproducibility and scalability.

A.2 Training microstructure samples

Figure 7 shows snapshots from a single time series within the training dataset, which contains 67 such series. The snapshots represent the evolution of phase separation during the 3D simulation of the Cahn-Hilliard equation, illustrating the dynamic changes in microstructures over time. The time series highlights the temporal progression and formation of distinct microstructures as the concentration field evolves. The Cahn-Hilliard model accounts for both thermodynamic forces and

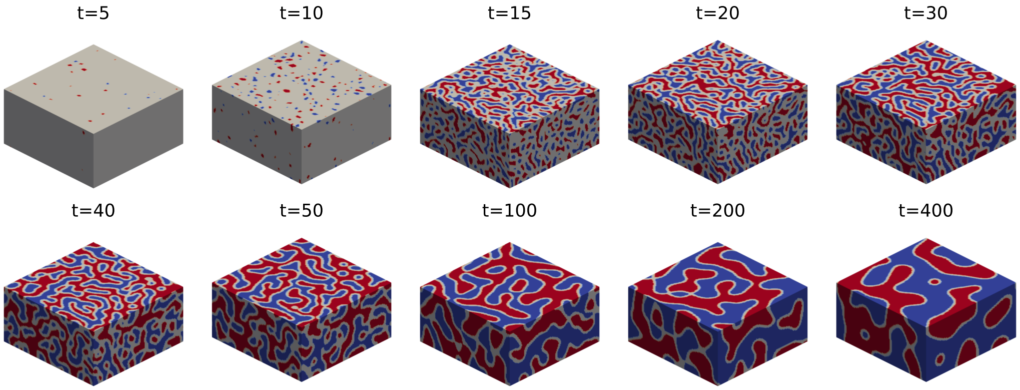
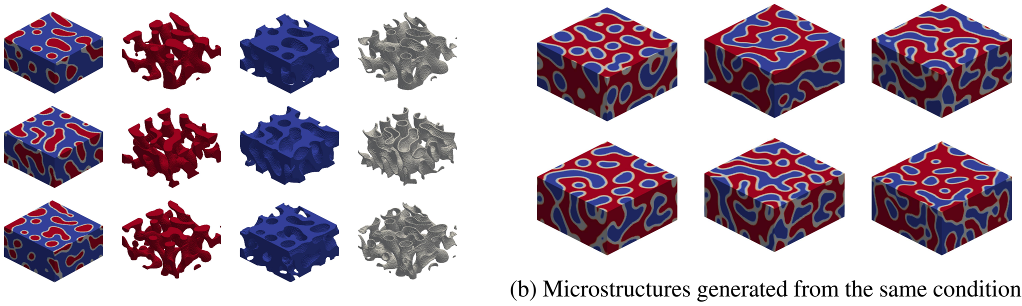


Figure 7: A sequence of 10 snapshots from one time series out of 67 in the entire dataset, illustrating the evolution of phase separation in a 3D simulation of the Cahn-Hilliard equation.

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 [49, 50].

A.3 Inference microstructure samples



(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 mix 0.3 and 0.2, respectively. Tortuosity of both phases is 0.3.

Figure 8: Three phase inference micsrostructure samples.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes] see section 4

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes] see section 2 and A.1

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No] Available to reviewers upon request; will be released publicly upon acceptance.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes] see section 2 and A.1

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No] Training the whole framework takes more than 10 days and we do not have the resources to train multiple models with the same hyperprameters.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes] See A.1

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [No] Code, data and model weights will be released publicly upon acceptance.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]