

Machine-learned domain partitioning for computationally efficient coupling of continuum and particle simulations of membrane fabrication

Matthias Busch^{a,*}, Gregor Häfner^{b,*}, Jiayu Xie^b, Marius Tacke^c, Marcus Müller^b, Christian J. Cyron^{a,c}, Roland C. Aydin^{a,c,*}

^a*Institute for Continuum and Material Mechanics, Technical University of Hamburg, Eißendorfer Straße 42, Hamburg, 21073, Germany*

^b*Institute for Theoretical Physics, Georg August University Göttingen, Friedrich-Hund-Platz 1, Göttingen, 37077, Germany*

^c*Institute of Material Systems Modeling, Helmholtz-Zentrum Hereon, Max-Planck-Straße, Geesthacht, 21502, Germany*

Abstract

All simulation approaches eventually face limits in computational scalability when applied to large spatiotemporal domains. This challenge becomes especially apparent in molecular-level particle simulations, where high spatial and temporal resolution leads to rapidly increasing computational demands. To overcome these limitations, hybrid methods that combine simulations with different levels of resolution offer a promising solution. In this context, we present a machine learning-based decision model that dynamically selects between simulation methods at runtime. The model is built around a multilayer perceptron (MLP) that predicts the expected discrepancy between particle and continuum simulation results, enabling the localized use of high-fidelity particle simulations only where they are expected to add value. This concurrent approach is applied to the simulation of membrane fabrication processes, where a particle simulation is coupled with a continuum model. This article describes the architecture of the decision model and its integration into the simulation workflow, enabling efficient, scalable, and adaptive

*Corresponding authors

Email addresses: `matthias.busch@tuhh.de` (Matthias Busch),
`gregor.haefner@uni-goettingen.de` (Gregor Häfner), `roland.aydin@hereon.de`
(Roland C. Aydin)

multiscale simulations.

1. Introduction

Fabrication of diblock copolymer membranes is a challenging process [1]. Computer simulations are a valuable tool to optimize this process. Such simulations can rely on different approaches. On the one hand, phase-field continuum models can be used. They are computationally efficient, but lack resolution of details on the molecular scale. On the other hand, particle simulations are possible. They capture many more details but are also computationally much more expensive. This limits their use in large-scale applications such as membrane fabrication, biological assemblies, or complex fluids in realistic geometries. To address this challenge, one can use multi-scale, multi-fidelity, and hybrid modeling techniques. Generally, these can couple fine-grained, high-fidelity representations (like particle models) and coarse-grained, low-fidelity representations (like continuum models), activating the former only where necessary, thereby increasing the global computational efficiency. Seminal methods include the heterogeneous multiscale method (HMM) [2, 3], the quasi-continuum (QC) approach [4], and coupled atomistic/continuum (CAC) schemes [5].

However, traditional coupling schemes typically use heuristic triggers – e.g., stress, strain or composition gradients – to switch fidelity, which can overlook subtle error excursions or over-resolve low-error regions. Machine-learning (ML) offers a data-driven alternative, enabling surrogate models to predict the local error between different simulation models and guide adaptive coupling. Surveys of multi-fidelity frameworks illustrate how surrogate-based gap estimation accelerates uncertainty quantification and optimization tasks by combining high- and low-fidelity data on the fly [6]. In turbulence modeling, Duraisamy et al. [7] review how ML-augmented subgrid closures can learn and quantify model-form error, demonstrating the value of data-driven discrepancy estimation for guiding selective high-fidelity computation. Advances in scientific ML – such as deep multi-fidelity Gaussian processes (GP) [8], nonlinear information fusion in GPs [9], and time-series ML-error models [10] – demonstrate the growing capability in learning and quantifying fidelity discrepancies.

Beyond methodological surveys, recent studies integrate ML directly into multiscale simulation infrastructures. Sanderse et al. [11] review the broad challenge of applying ML to multiscale closure problems, emphasizing the

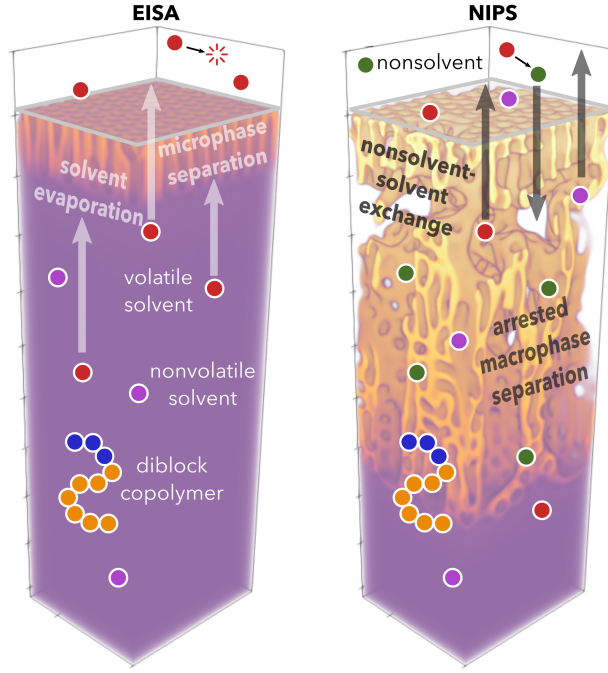


Figure 1: Schematic representation of the evaporation-induced self-assembly (EISA) and nonsolvent-induced phase separation (NIPS) process. During EISA the volatile solvent evaporates from the casting solution – diblock copolymers dissolved in a two-solvents mixture – by diffusion. The evaporation is captured in simulation by conversion of solvent into the gas phase at the top. During NIPS nonsolvent-solvent exchange is initiated by exchanging the gas phase for nonsolvent. Inside the solution film, a combination of macrophase separation and simultaneous polymer vitrification results in the final nonequilibrium structure.

need for physics-aware surrogate forms and discretization invariance. Nguyen et al. [12] discuss opportunities and challenges in ML-augmented multi-scale modeling, from data scarcity to interpretability in engineering systems. Coupling ML surrogates with high-performance frameworks has enabled dynamic-importance sampling to select scales adaptively in real time [13]. Additionally, Benson et al. [14] report an ML-driven scale-bridging infrastructure for active matter simulations, which automatically learns mapping operators between fine- and coarse-scale representations.

In this work, we focus on the question of how to use machine learning to efficiently couple models with a different degree of details for simulating the fabrication of integral-asymmetric, isoporous diblock-copolymer membranes

[15, 16, 17, 18, 19]. These are cast from a diblock copolymer solution and consist of an isoporous top layer, resulting in ideal selectivity, and underneath a coarse, sponge-like substructure, giving the membrane mechanical stability while retaining permeability. The fabrication process – solvent evaporation and nonsolvent-induced phase separation (SNIPS) – involves two steps: After casting a block copolymer solution onto the substrate with a doctor blade, evaporation-induced self-assembly (EISA) occurs – a volatile solvent evaporates from the film leading to the self-assembly of the diblock copolymer into a cylindrical morphology that is oriented perpendicularly to the film surface. In a second step, the liquid film is immersed in a coagulation bath – commonly a nonsolvent such as water – thereby inducing nonsolvent-induced phase separation (NIPS). In this step, the nonsolvent penetrates the film through the cylindrical phase, leading to the formation of open pores, while at the same time, the polymer vitrifies, freezing the nonequilibrium structure, as the solvent penetrates into the film. Underneath the selective, isoporous top layer, the nonsolvent macrophase separates from the polymer, which results in the coarse sub-structure. The SNIPS process is depicted in Figure 1, showing how the problem is tackled by simulations. Achieving the ideal membrane morphology via this process is challenging because it depends on a diverse set of material and process parameters. Simulations can give valuable insights into the physics at play, as well as guide experiments by revealing parameter dependencies. With modern supercomputing, continuum and particle simulations are able to capture the relevant length and time scales, and have provided valuable insights into the EISA process [20, 21, 22, 23], the NIPS process [24, 25], as well as the full SNIPS process [26, 27] based on particle simulations. Nonetheless, the latter simulations are computationally demanding and large-scale parameter studies are challenging with such techniques. Continuum modeling offers a computationally more efficient approach to modeling polymer self-assembly and the SNIPS process. Using the GPU-accelerated Uneyama-Doi model (UDM)-software [28, 29], we demonstrate that continuum modeling allows modeling the full SNIPS process and results in qualitatively similar results as the coarse-grained particle simulations; nonetheless, quantitative differences remain.

Planning in the future to combine the two models in a multi-fidelity modeling approach, in this work, we present an MLP-based decision model that predicts, at runtime, the local divergence between particle and continuum simulations. By triggering high-fidelity simulation only where the predicted error exceeds a tolerance, a concurrent hybrid solver can be expected to

achieve significant speedups while controlling accuracy – offering a blueprint for adaptive multiscale coupling in other domains.

2. Simulation of membrane fabrication

To model the two-step fabrication process, we consider a system of a cylinder-forming AB diblock copolymer (typically poly(4-vinylpyridine) (P4VP)-polystyrene (PS)) dissolved in a volatile solvent (THF) and a non-volatile one (Tetrahydrofuran (DMF)). For the EISA-process the gas phase is modeled as a phase-separated liquid phase, while this species is exchanged for non-solvent, water, at the start of the NIPS-process. In the following, the different species are denoted A (P4VP), B (PS), S (THF), C (DMF), G (gas) and N (nonsolvent). The two simulation schemes are shortly introduced in the following, while a more detailed description of the two is given in Appendix A.

2.1. Particle-based simulations

The particle simulations employ a soft, coarse-grained model [30, 31] that represents several monomer repeat units by a single particle. The Hamiltonian is split into strong bonded (b) and weak non-bonded (nb) interactions, $\mathcal{H} = \mathcal{H}_b + \mathcal{H}_{nb}$. While the bonded interactions – harmonic springs – are treated exactly, the non-bonded interactions are accurately replaced by interactions with fluctuating external fields. For this, the local normalized concentration fields, $\phi_\alpha(\mathbf{r})$, for $\alpha = A, B, S, C, G, N$ and space $\mathbf{r}$, and the external fields are evaluated as a function of these. The single-chain-in-mean-field (SCMF)-algorithm is employed, which transiently fixes the external fields on time scales at which they evolve only marginally and therefore decouples the molecules for a short time period. This allows for massive parallelization on modern supercomputer architectures, which is efficiently implemented in the software SOft coarse-grained Monte carlo Acceleration (SOMA) [32]. In the software, particle positions are propagated in time by smart-Monte-Carlo moves, resulting in Rouse-like dynamics of the polymer chains [33]. Hence, it is capable of modeling the non-equilibrium time evolution of the membrane fabrication process.

In the particle model, N_0 refers to the chain-contour discretization of the polymer and lengths are measured in units of R_e , the mean end-to-end distance of the ideal polymer chain. Time is measured in units of time, τ_R , the time it takes the polymer to diffuse its own mean end-to-end distance.

The density of the system is set by the invariant degree of polymerization, $\bar{N}$, which controls the strength of thermal fluctuations. Interactions are modeled as local binary repulsions between distinct species, in the form of the Flory-Huggins parameters $\chi_{\alpha\beta}$ for $\alpha, \beta = A, B, S, C, G, N$. To model the vitrification of the polymer in the absence of plasticizing solvent, the inverse segment friction is modified depending on the local composition, $m_\alpha(\{\phi_\beta\})$. It follows a sigmoidal curve that has unity at vanishing polymer concentration and vanishes for high ones, with the turning point given by a threshold polymer concentration, ϕ_P^* .

2.2. Continuum simulations

Within the continuum model, the normalized concentration fields, $\phi_\alpha(\mathbf{r})$, fully determine the system's state, and are directly propagated in time via Ginzburg-Landau-type dynamics, also known as model-B, [34]

$$\frac{\partial \phi_\alpha(\mathbf{r}, t)}{\partial t} = \nabla \cdot \left[R_e^2 \lambda_\alpha(\{\phi_\beta(\mathbf{r})\}) \phi_\alpha(\mathbf{r}) \frac{\nabla \mu_\alpha(\mathbf{r}) R_e^3}{\sqrt{\bar{N}} k_B T} \right] + s_\alpha(\mathbf{r}), \quad (1)$$

where k_B is the Boltzmann constant and T the temperature. The chemical potentials, $\mu_\alpha(\mathbf{r})$ are derived from the free-energy functional, $\mathcal{F}[\{\phi_\alpha\}]$, that was proposed by Uneyama and Doi for arbitrary blends of copolymers and homopolymers or solvents [35, 36]. The term $s_\alpha(\mathbf{r})$ denotes local sinks and sources that are used to model the evaporation process during EISA[23], as well as the solvent-nonsolvent exchange during NIPS.

The flexibility of the theory allows treating a wide range of systems and phenomena. In addition, the equilibrium phase behavior, as well as the nonequilibrium dynamics, shows great similarity to more detailed theories like self-consistent field theory (SCFT) for polymers [37, 35] or the above-described particle model for a number of systems [23, 28, 29]. In Equation 1, λ_α determines the mobility of each species and its variation with polymer concentration models the polymer vitrification. For this, one can simply identify the inverse segmental friction of the particle model to be $m_\alpha(\{\phi_\beta\}) = \lambda_\alpha(\{\phi_\beta\})/\lambda_0$, where λ_0^{-1} sets the time scale in the continuum model.

The model implementation utilizes the UDM software framework [28, 29], which offers a CUDA/C-based GPU implementation of the aforementioned model. To model the SNIPS process, including the required polymer vitrification, the software has been extended to feature density-dependent mobility.

2.3. Choice of model parameters

Apart from the time scale, both the continuum and the particle model operate on the same set of parameters, which enables an excellent comparison between the two models. As the base parameter set of this work, we employ the same parameters as established in the work of Blagojevic *et al.* [27], for which the SNIPS process results in an ideal membrane morphology within the particle simulations. Deviations from the base parameter set will be specifically denoted. Most importantly, the interactions are varied for the generation of training data. The process and material as well as discretization parameters are chosen as follows.

In the particle model, the chain-contour discretization of the diblock copolymer is chosen $N_0 = 64$, while solvent molecules are modeled as oligomers of $N_\alpha = 8$ for $\alpha = S, C, G, N$. In the continuum model, only the ratio of the two plays a role which is kept the same. The block ratio of the diblock copolymer is chosen $f_A = 0.3125$, and in the particle model, we choose $\sqrt{N} = 380$. The symmetric matrix of Flory-Huggins parameters for the reference system reads

$$\chi N_0 = \begin{pmatrix} & A & B & S & C & G & N \\ A & 0 & 30 & 15 & 0 & 200 & 10 \\ B & 30 & 0 & 0 & 3 & 140 & 150 \\ S & 15 & 0 & 0 & -35 & 10 & -30 \\ C & 0 & 3 & -35 & 0 & 120 & -30 \\ G & 200 & 140 & 10 & 120 & 0 & - \\ N & 10 & 150 & -30 & -30 & - & 0 \end{pmatrix}. \quad (2)$$

These are inspired by experimental findings and chosen such that the volatile solvent, S , is selective for the matrix-forming block B , whereas the non-volatile solvent, C , prefers the cylinder-forming block, A . The nonsolvent, N , is highly incompatible with the polymer, particularly with the matrix-forming block, B , leading to macrophase separation upon penetration into the solution film. For the generation of training data, we wish to cover a broad range of scenarios within the SNIPS process. Therefore, we randomly vary this interaction matrix, as will be described in detail below.

Finally, to make the simulations fully comparable, the time scales, λ_0^{-1} in the continuum model and $\tau_R = 31\,355\text{MCS}$ (Monte-Carlo steps) in the particle model need to be matched. For this study, the ratio of $\tau_R\lambda_0 = 1.3125$

was determined by minimizing the deviation of concentration fields during the EISA process.

In the continuum model, a time step of $\Delta t = 2 \cdot 10^{-5} \lambda_0^{-1}$ is chosen to propagate the system in time.

The parametrization of the species mobility is chosen, such that the threshold polymer concentration is $\phi_P^* \approx 0.72$. The initial casting solution is disordered with equal concentrations of polymer and the two solvents $\phi_P(t=0) = \phi_S(t=0) = \phi_C(t=0) = \frac{1}{3}$. For the introduction of the modeling, we choose domains of size $V = 14 \times 16 \times 51.2 R_e^3$, which is discretized into a collocation grid with spacing $\Delta x = \frac{R_e}{10}$. For the generation of training data, a smaller domain, with size $V = 5.6 \times 6.4 \times 51.2 R_e^3$, was chosen. In all cases, the domain has periodic boundary conditions in x - and y -direction, and a no-flux boundary condition in z -direction. This is achieved via reflective boundary conditions in the continuum model, and in the particle model via a wall, which particles cannot penetrate. The process time for EISA and NIPS are chosen equal for both, $T_{\text{EISA/NIPS}} = 16\tau_R = 21\lambda_0^{-1}$ in the particle and continuum simulations respectively.

2.4. Example simulations

With this setup, we are able to model the SNIPS process with both simulation schemes. We present the complete SNIPS process for both simulation approaches using the reference system parameters. Figure 2 shows 3D concentration fields of the matrix-forming block B during membrane formation at different times for the two simulation schemes, as well as the final concentration field of the nonsolvent, $\phi_N(\mathbf{r})$.

During EISA ($t < 16\tau_R$ in the particle model, $t < 21\lambda_0^{-1}$ in the continuum model), the volatile solvent, S , evaporates, causing the film surface to retract and the polymer to enrich at the interface. This increased polymer concentration at the surface – skin formation – promotes its self-assembly into hexagonally ordered cylinders perpendicular to the film surface, forming a well-ordered layer. After the solvent-evaporation step, this layer exhibits a thickness of approximately $4R_e$. These dynamics are equivalent in both simulation schemes.

Upon the next process step, at time $t = t_{\text{EISA}} = 16\tau_R = 21\lambda_0^{-1}$, NIPS initiates through an instantaneous conversion of gas to nonsolvent. The nonsolvent, N , being incompatible with polymer but miscible with solvents S and C , penetrates the polymer skin primarily through the A-rich cylinder cores due to preferential compatibility. As remaining solvent is displaced,

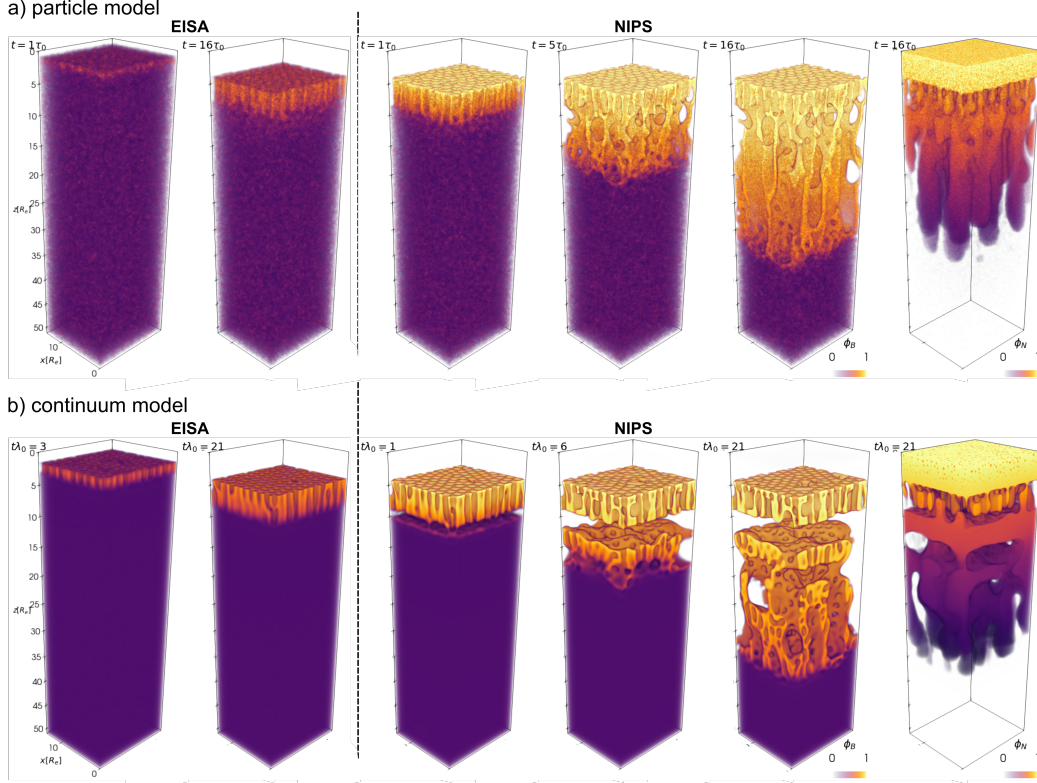


Figure 2: Time evolution of the B-block concentration fields, $\phi_B(\mathbf{r})$, during the EISA and NIPS process in a) the particle simulations and b) the continuum simulation. Indicated times are measured from the start of each process. The far right panel additionally shows the nonsolvent concentration field, $\phi_N(\mathbf{r})$, after the full SNIPS process.

the polymer concentration exceeds the vitrification threshold ϕ_P^* , arresting the self-assembled structure in a glassy state.

The thermodynamics underpinning structure formation during EISA and NIPS can be understood from the spinodal curves of membrane casting solutions at different nonsolvent concentrations [27], shown on the Gibbs triangle in Figure 3(a). The spinodal, evaluated using the random-phase approximation (RPA)[38, 39], marks the loss of linear stability of the spatially homogeneous state against composition fluctuations with either a finite (dashed lines) or infinite (solid lines) characteristic length scale. As the RPA is a standard approach for determining the spinodal in multicomponent polymer systems, we refer the reader to the literature for technical details [39, 40, 41, 42]. The spinodal curve with $\phi_N = 0$ delineates the instability

boundary of the casting solution used in EISA. At $t = 0$, the initial concentrations, $\phi_P(t = 0) = \phi_S(t = 0) = \phi_C(t = 0) = \frac{1}{3}$, lie within the (meta)stable region of the Gibbs triangle. As solvent S evaporates, ϕ_S decreases near the solution surface while ϕ_P and ϕ_C increase, shifting the state point of this layer toward the spinodal (black dashed curve in Figure 3(a)). Upon crossing the spinodal, spontaneous phase separation occurs at a finite length scale, giving rise to the vertically arranged cylinders observed in Figure 2.

During NIPS, nonsolvent molecules penetrate the film as solvent and nonsolvent exchange. As shown in Figure 3(a), the (meta)stable region of the solution shrinks with increasing ϕ_N , driven by the strong polymer–nonsolvent incompatibility. Notably, once ϕ_N exceeds a threshold of about $\phi_N^* = 0.0033$, a segment of the spinodal curve corresponding to instability against fluctuations of infinite length scale emerges (solid line). This solid instability boundary separates the dashed boundary into two branches and intersects them at two points, i.e., the Lifshitz points [42, 43]. In contrast to the dashed portions of the spinodal curve, the solid segment expands much more rapidly with increasing ϕ_N and dominates the entire spinodal for $\phi_N \gtrsim 0.004$. When the state point of the casting solution crosses this solid boundary, phase separation between polymer and nonsolvent proceeds on a macroscopic length scale, giving rise to the macroporous structure observed in Figure 2.

The two simulation approaches exhibit qualitatively different behavior during NIPS. In the particle simulations, nonsolvent penetration induces controlled macrophase separation, leading to the formation of a sponge-like substructure with moderate-sized macrovoids connected to the cylindrical top layer. In contrast, the continuum model predicts more pronounced macrophase separation upon nonsolvent penetration, resulting in the formation of large macrovoids spanning the entire lateral simulation domain. This leads to complete detachment of the sponge-like substructure from the cylindrical open-pore top layer, creating a distinct interface between these regions. The continuum model’s prediction of complete layer detachment represents a more extreme manifestation of the phase separation instabilities, while the particle-based approach captures the intermediate connectivity observed experimentally in many SNIPS membranes.

This difference of dynamics during the NIPS process can also be observed in Figure 3 b) and c), which show the laterally averaged concentration profiles of each species, as well as a 2D slice of the concentration difference between the two polymer blocks, $\phi_A - \phi_B$, at different times during the SNIPS process in both models. In each panel, the top row shows the base parameter combi-

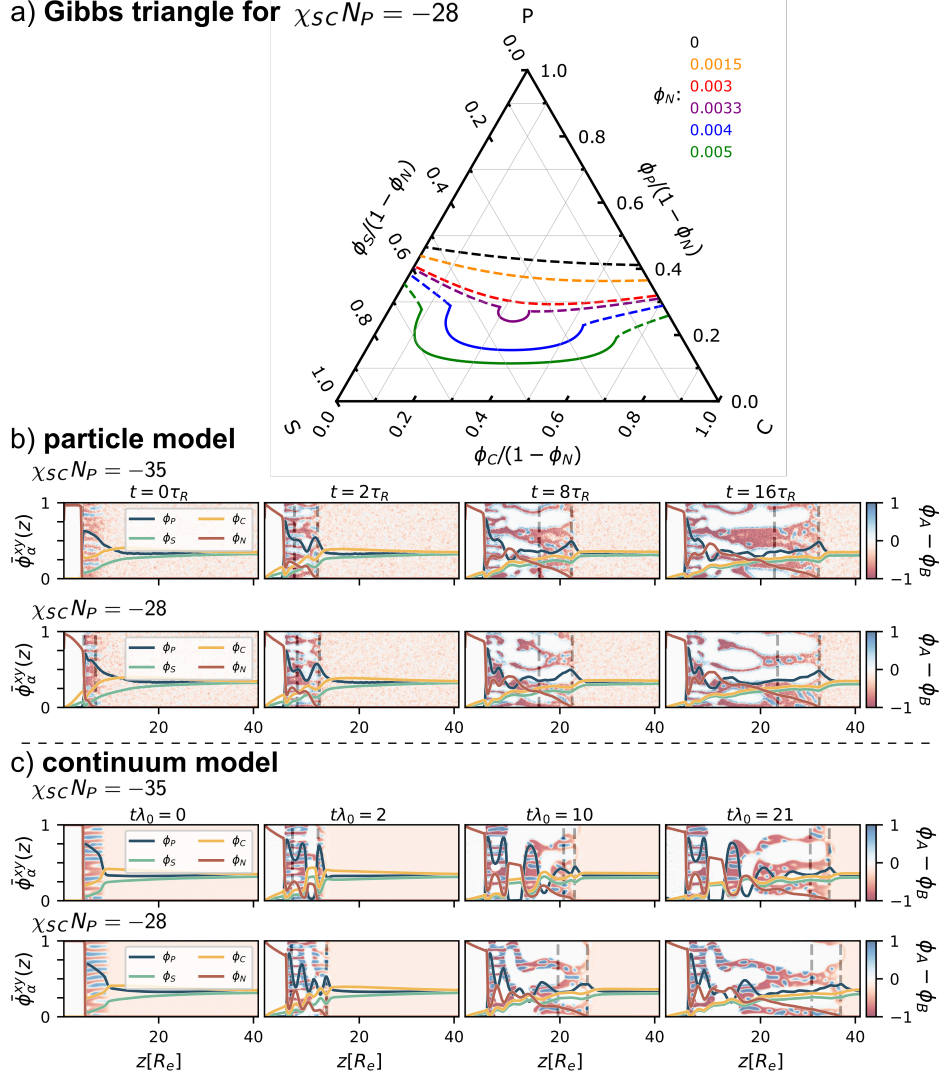


Figure 3: Panel a) shows spinodal curves of membrane casting solutions at different nonsolvent concentrations on the Gibbs triangle. Dashed and solid lines indicate instabilities of the spatially homogeneous state against composition fluctuations with finite and infinite length scales, respectively. Panels b) and c) show the 1D laterally-averaged concentration profiles, $\bar{\phi}_\alpha^{xy}(z)$ (lines) and 2D cross-sections of the block concentration difference, $\phi_A(\mathbf{r}) - \phi_B(\mathbf{r})$ (background) for a) the particle simulations and b) the continuum simulations of the base parameter pair (top) and the weakened solvent attraction, $\chi_{SC}N_P = -28$ (bottom). Time increases from left to right as indicated. For the 2D cross sections, the y -axis is scaled the same as the z -axis. The approximated front positions are indicated as dashed gray lines – vitrification front left of the structure-formation front.

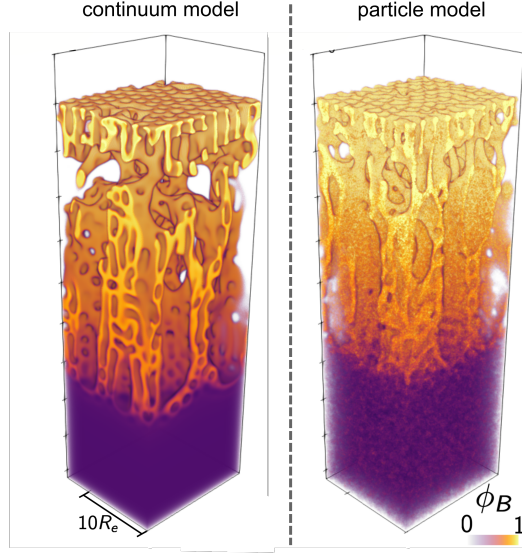


Figure 4: Final B -block concentration field, $\phi_B(\mathbf{r})$ after the NIPS process in the continuum simulation (left) and in the particle simulation (right) for the reduced solvent attraction, $\chi_{SC}N_P = -28$, compared to the base parameter combination.

nation. The detachment of the cylindrical phase from the substructure in the continuum simulation expresses itself in a vanishing polymer concentration profile at $z \approx 10R_e$. In contrast, the polymer concentration stays finite in this region in the particle simulation. Nonetheless, a dip of this concentration is also visible.

A minor change to the base parameter combination changes the membrane morphology qualitatively in the continuum simulation, resulting in the connectivity of the two membrane regions. This occurs when the S - and C -solvent attraction that leads to a depletion of solvents in the polymer skin, is decreased, $\chi_{SC}N_0 = -28$. This scenario is plotted in the bottom row of each panel in Figure 3 b) and c), which shows that the decreased attraction leads to weaker concentration gradients inside of the cylindrical domains after EISA, so that the nonsolvent penetration can be expected to be slower after initiation of NIPS. Additionally, the less pronounced increase of solvent concentration underneath the polymer skin provides a reduced pre-pattern for the macroscopic nonsolvent phase to form in this region. In contrast, in the particle simulation the change in material parameters makes no qualitative difference.

The final 3D B -block concentration field after NIPS for the solvent attraction is shown in Figure 4, showing the polymer structure has a hexagonal top-layer that is connected to the sponge-like substructure underneath in both models. Hence, the continuum simulations are qualitatively giving similar results, with variations in the exact parameters values, but need to be adjusted to give a full quantitative overlap with the more-accurate particle simulations.

In addition to the morphological analysis, the two simulation approaches reveal a three-layer structure during the later stages of NIPS ($t > 36\tau_R$), that effectively describes a 1D problem:

Layer 1: Arrested phase separation. This vitrified region extends from the hexagonal polymer skin at the film surface into the upper macroporous substructure. Here, low solvent concentrations in polymer-rich domains cause vitrification of the matrix-forming block B upon crossing the glass-transition threshold. Dynamics consist primarily of nonsolvent-solvent exchange through the static, complex porous geometry.

Layer 2: Active structure formation. This region extends from the vitrification front to the structure-formation front, where nonsolvent concentration sufficiently high to induce macrophase separation and/or solvent depletion triggers microphase separation. The macro- and microphase separation fronts coincide for the chosen SNIPS parameters, resulting in polymer concentration increase and A-core micelle formation.

Layer 3: Homogeneous solution. Deep within the film, the solution remains laterally homogeneous without phase separation. Only shallow concentration gradients exist.

The transitions between these three layers, the vitrification front and the structure-formation front, are approximated for all simulations and indicated in Figure 3. To approximate the vitrification front, we take the position where laterally-averaged absolute difference of the current concentration with the future concentration, $\Delta\tau = 4.2\lambda_0^{-1} = 3.2\tau_R$, falls below a 30% threshold of the maximal change. Hence this quantifies the region in which the future concentration change is sufficiently small. This results in the left gray line of the panels in Figure 3 b) and c). To approximate the structure-formation front, we use the Gibbs triangle of Figure 3 a), and identify the lowest nonsolvent concentration for which macrophase separation occurs to be $\phi_N = 0.0033$. Hence, the structure-formation front occurs at the largest z -value, where the laterally-averaged nonsolvent concentration crosses this critical concentrations, and is observable as the right gray line in the panels of Figure 3 b)

and c). With these fronts indicated, the above-described three layers become clearly visible.

Improvement of the continuum simulations most likely target layer 2, where the structure formation occurs. These are therefore spatially confined, hinting towards our planned strategy of coupling the full continuum simulations to locally-confined, and therefore computationally more efficient, particle simulations.

2.5. Strategy for efficient SNIPS simulations

Given the qualitative similarity of the continuum model results compared with the more accurate particle simulations, the continuum simulations appears ideal to perform high-throughput parameter studies of the SNIPS process. While the qualitative similarity is given, there are clear quantitative differences that need to be overcome.

The above-described three-layer model provides a strategy for using both models to perform a computationally 'efficient, yet accurate simulation. The main differences between the two simulation schemes can be expected to occur in the second layer – the structure formation zone. Performing a small-scale particle simulation in this narrow zone, across the full lateral dimensions, while modeling the full domain with the continuum model, would provide the required accuracy during the structure formation. Such a scheme remains computationally efficient because the particle simulation is only performed on a small sub-domain.

However, identifying the size and position of the sub-domain requires *a-priori* knowledge, where the continuum model deviates most from the particle model. In the following, we demonstrate that this can be achieved through a MLP-based prediction of the laterally-averaged error. The choice of sub-domain can then be performed by a simple post-processing of the live-performed error prediction. This ML-based decision model should work reliably for a wide range of parameters, preferably even outside of the range of the training data. Here, we can make use of the fact that the SNIPS process is expected to follow the above three-layer model. Therefore, the ML-model should be able to predict the ideal sub-domain even if it is only trained on a finite range of interaction parameters.

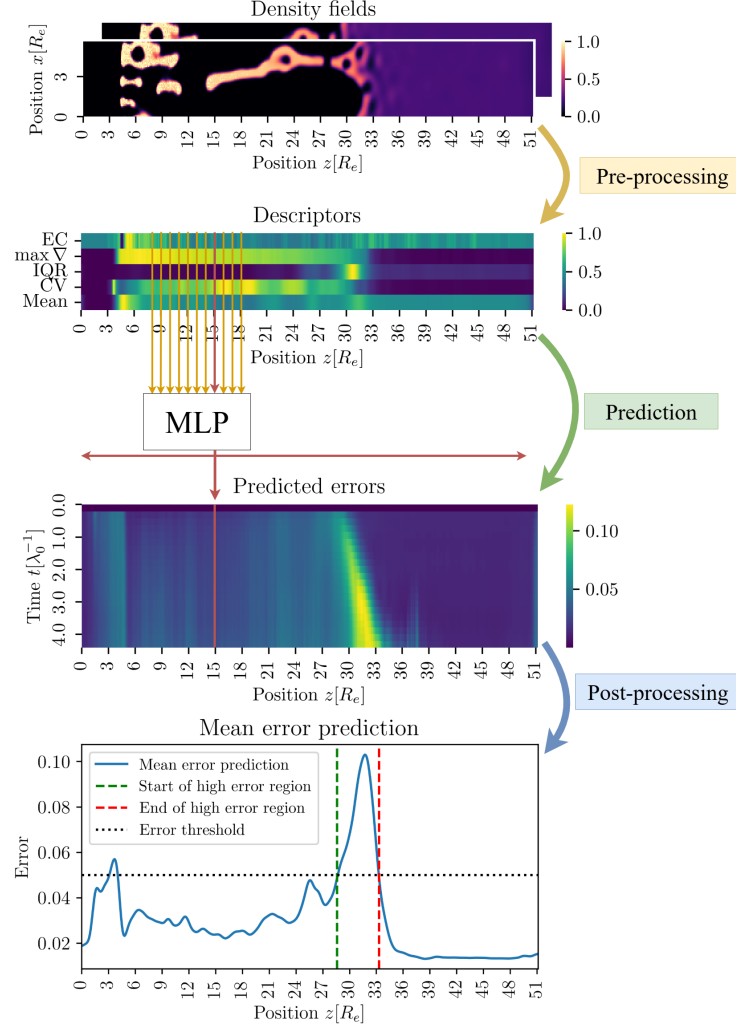


Figure 5: Overview diagram for the decision model. The different steps of data processing with intermediate results are shown. For details on each step, refer to the following sections 3.1, 3.2 and 3.3. In the pre-processing step descriptors are extracted from the density field (5 for each z -layer, abbreviations: CV – coefficient of variation; IQR – interquartile range of finite differences; max ∇ – maximum gradient magnitude; Mean – layer mean; EC – Euler characteristic), with the IQR being computed from the finite difference of two consecutive (in time) density fields. The prediction is done by a MLP, which takes descriptors of multiple z -layers as input and outputs predicted errors for a single z -layer at multiple future time steps. The post-processing aggregates the errors for the requested time range and finds the largest error region above a threshold.

3. Decision-model architecture

An overview of the architecture of the decision model for the NIPS-process is shown in Figure 5. The model has to start with a raw density field from the continuum simulation and arrive at a partitioning for the domain in the end. The process consists of the following three steps:

- **Pre-processing (section 3.1):** The raw data, the B-block concentration field $\phi_B(\mathbf{r})$, is smoothed with a Gaussian filter and information-dense descriptors are extracted. The descriptors reduce the dimensionality of the data by two, such that only the vertical dimension, z , remains.
- **Prediction (section 3.2):** A trained machine learning model, an multilayer perceptron (MLP), estimates the time evolution of z -resolved divergence between both simulation methods.
- **Post-processing (section 3.3):** The divergence is interpreted as the local error, where a large divergence means that the continuum simulation is not an accurate approximation and the particle simulation needs to be applied.

This architecture aims at providing an efficient, flexible, and reliable decision model to partition the domain based on an error prediction. The flexibility denotes the variability of domain size in all three dimensions, independence from simulation time, as well as the ability to predict the divergence for a broad range of parameter combinations. Variations of the latter may change the behavior of the simulation significantly. Within all scenarios, the model needs to reliably predict at least qualitatively correct spatiotemporal regions in which the continuum simulations are performing at an accuracy below a prescribed threshold. In the next sections, details of the architecture will be explained. In this work, we focus on the NIPS process. However, the procedure is the same for the EISA process and the resulting MLP differs only in the selection of descriptors and the input phase.

3.1. Pre-processing

The aim of the pre-processing is to provide reliable, information-dense input to the machine learning model while reducing data complexity. In our case, the steps include Gaussian smoothing, finite-difference computation, descriptor extraction, and scaling.

Raw data

The concentration fields of the five different species contain mostly redundant information. To increase efficiency, only the B-block concentration field, corresponding to the matrix-forming membrane material, is used as input. Hence, the raw data input consists of the B-block concentration fields at two consecutive time steps, $\phi_B(\mathbf{r}, t)$ and $\phi_B(\mathbf{r}, t - 0.21\lambda_0^{-1})$. The two consecutive time steps are used to calculate the finite temporal difference, providing both the current density field and its temporal change as inputs for the descriptor calculation (Figure 5).

Descriptors

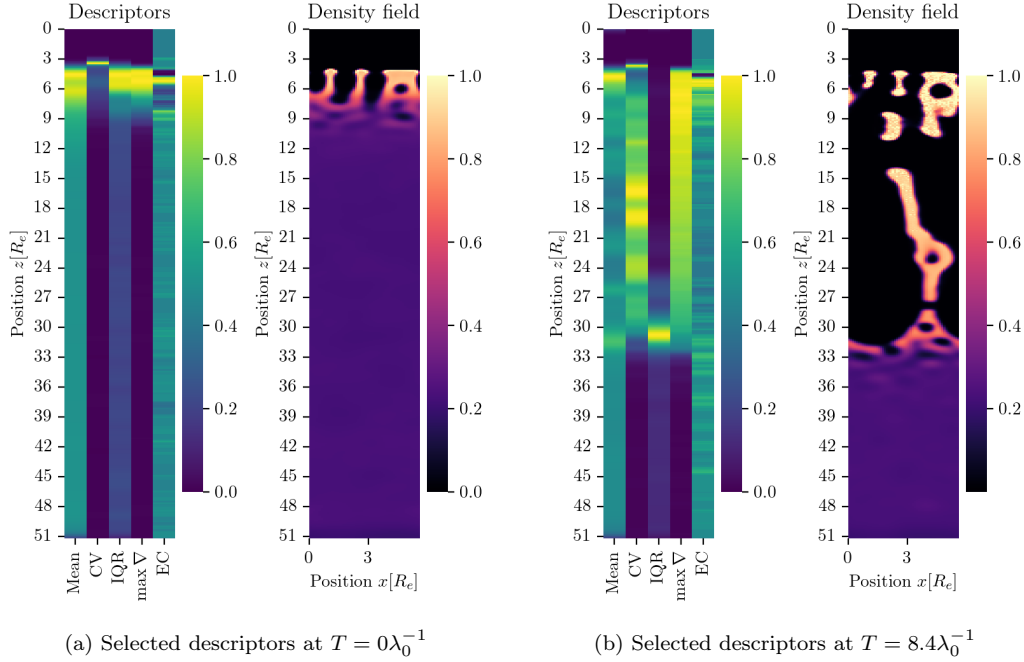


Figure 6: Heatmaps showing the descriptors selected by the greedy feature selection compared to a 2D slice through the density field, $\phi_B(\mathbf{r})$, they were extracted from. On the x -axis the selected descriptors (abbreviations: CV – coefficient of variation; IQR – interquartile range of finite differences; max ∇ – maximum gradient magnitude; Mean – layer mean; EC – Euler characteristic) and the x position are shown, while the y -axis depicts the z -position and is shared between all diagrams. The descriptors are calculated for each z -layer of the density field.

A total of 63 descriptors were computed for each z -layer (spacing $0.1R_e$) of the density field, $\phi_B(\mathbf{r})$, and its temporal finite difference in every sample,

including statistical measures (e.g., mean, variance, percentiles) and image-based features that treat the density field as a grayscale image [44]. From these, a greedy feature selection algorithm identified the five best performing features. It started with a single feature and iteratively chose the respective best performing next feature. After five features, the improvements became negligible, and the contribution of the fifth feature was already unclear (see Figure 11).

The five best performing descriptors for prediction are the coefficient of variation (CV), the interquartile range (IQR) of finite differences, the maximum gradient magnitude, the mean, and the Euler characteristic. This section defines these descriptors, explains how they are computed, and describes their physical interpretation.

Mean μ

The mean μ of the layer describes changes in the structure, similar to the CV and marks the beginning and the end of the structure. However, compared to most other descriptors, there is no significant decrease to zero after the structure-formation front:

$$\mu = \frac{1}{N_x N_y} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} \phi_B(i\Delta x, j\Delta y, z) \quad (3)$$

where N_x, N_y are the number of grid cells in x - and y -direction.

Coefficient of variation (CV)

The CV of the 2D layer correlates strongly with the formed structures in the domain and is defined as

$$\text{CV} = \frac{\sigma_{\text{layer}}(z)}{|\mu_{\text{layer}}(z)| + \epsilon} \quad (4)$$

where $\mu_{\text{layer}}(z)$ and $\sigma_{\text{layer}}(z)$ refer to the laterally (over x - and y -dimension) averaged mean and standard deviation of the concentration field $\phi_B(\mathbf{r})$. $\epsilon = 10^{-10}$ is needed to avoid division by zero.

Interquartile range (IQR) of finite difference

The IQR of the finite difference layer measures the spread of the middle 50% of values. Because it is calculated from the finite difference of two consecutive (in time) fields, it correlates strongly with the position of

the structure-formation front. It exhibits a single peak at this front and is close to zero elsewhere.

$$\text{IQR} = Q_{0.75} - Q_{0.25} \quad (5)$$

where $Q_{0.75}$ and $Q_{0.25}$ are the 75th and 25th percentiles of the concentration-field finite difference. The percentile Q_p represents the value below which $p\%$ of the data points lie, providing a general measure of the data distribution.

Maximum gradient magnitude ($\max \nabla$)

The maximum gradient magnitude in the layer indicates structures. This feature provides a very smooth transition from the top of the solution film to the structure-formation front. With this behavior it provides a reliable indicator of the position within the structure.

$$\max \nabla = \max_{xy} \left(\sqrt{\left(\frac{\partial \phi_B(\mathbf{r})}{\partial x} \right)^2 + \left(\frac{\partial \phi_B(\mathbf{r})}{\partial y} \right)^2} \right) \quad (6)$$

Euler characteristic (EC)

The Euler characteristic describes the difference between the number of components and the number of holes in the binarized layer. The threshold for binarization is at $\min_{xy}(\phi_B) + 0.75(\max_{xy}(\phi_B) - \min_{xy}(\phi_B))$. This descriptor shows deviations over the built structure. However, because it is very noisy, its benefit to the prediction remains unclear.

$$\text{EC} = \frac{1}{L_x L_y} (C_{0.75} - H_{0.75}) \quad (7)$$

where $C_{0.75}$ is the number of connected components and $H_{0.75}$ is the number of holes in the binarized layer. L_x and L_y denote the size of the layer and are used for scaling the descriptor.

Figure 6 shows the spatial distribution of the five selected descriptors for an example simulation at two time steps, alongside the corresponding density field slices. Descriptors are plotted along the z -axis, with the descriptor index and x -position on the abscissa. This representation illustrates how different descriptors capture distinct aspects of the evolving density field. All descriptors are domain size independent, meaning that they do not need

to be scaled with varying domain sizes. Specifically in the case of the Euler characteristic, the holes in the domain are counted, scaling linearly with the cross-sectional area, $L_x L_y$, of the domain, which justifies inversely scaling with it.

3.2. Error Prediction

Machine learning model

The goal of the machine-learning model is to predict the layer errors for multiple time steps. With the layer being indexed by l and the time being t , the error is defined as the mean error of the respective layer:

$$\text{Error}(t, l) = \frac{1}{N_x N_y} \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} |\phi_B^{\text{UDM}}(\mathbf{r}_{ijl}, t) - \phi_B^{\text{SOMA}}(\mathbf{r}_{ijl}, t)| \quad (8)$$

This task is satisfactorily achieved by an multilayer perceptron (MLP), which was implemented using scikit-learn [45]. It provides an easy-to-use, reliable, and efficient implementation of a neural network with all necessary flexibility for our use case. The MLP’s error prediction process, which it does on a layer-by-layer basis, is visualized in Figure 5. It predicts the errors of the next 21 time steps at once via its 21 outputs. Iterating the MLP through the whole z -dimension, one obtains a heatmap as shown in Figure 5. It shows an estimate for the future error at each layer for 21 time steps, based on which the partitioning can be done. As input, the MLP gets the descriptors of 11 layers, $1R_e$ spaced apart. Three of them are in front of the current prediction layer. All the hyperparameters of the MLP were optimized using the platform Weights and Biases. In total, the MLP has around 65000 parameters.

Model assumptions and design rationale

The central design choice is to formulate the prediction task on a per-layer basis rather than using the full domain as input. This is motivated by three main considerations: (i) independence of the model performance from the domain size in z -direction, (ii) enforced generalization across all layers, and (iii) a substantial increase in the number of training samples. The key assumption underlying this strategy is that the propagation of the structure-formation front – the dominant source of divergence between simulation methods – follows qualitatively similar dynamics throughout the simulation.

Compared to a global-domain architecture, the layer-wise formulation prevents the network from memorizing specific time-position patterns. Instead, it compels the MLP to treat each layer consistently, irrespective of

absolute position or simulation time. As a result, each occurrence of the structure-formation front is recognized and modeled in a similar manner.

This enforced invariance is particularly important for extending the decision model beyond the conditions present in the training set. Since training data must be generated on relatively small domains and over limited time horizons, the ability to generalize to longer domains and later time steps is essential. The layer-wise prediction not only removes the explicit dependence on the z -domain size, but also provides the necessary robustness for applying the model to simulations with larger spatial and temporal extents.

Production of training data

The application of the here-described ML-model is to couple the full continuum simulations of the SNIPS process to a locally confined particle simulation that overwrites the continuum model in a region where the latter is inaccurate. Hence, the goal of the ML-model is to predict the error of the continuum simulation compared to the particle simulation, with the goal of determining the region in which it is least accurate to choose as a sub-domain for the particle simulation.

Optimal training data would mirror exactly the setting of the application. This would involve simulating a range of fully coupled simulations, with randomly partitioned domains, to include all sources of errors between the particle solution and the continuum solution. This would introduce orders of magnitude of additional complexity and computational cost not only for prediction, but also for training data generation. Thus, we need to simplify the problem with the following assumptions:

- The time evolution of a subdomain is relatively independent of its coupling. This means, for all of our scenarios, a subdomain of a particle simulation embedded into a continuum simulation behaves approximately like the same simulation being part of a full particle simulation.
- The characteristics of the simulation are from a certain point onwards similar enough to be able to generalize to the later time steps.
- Because of the reduction to descriptors, the MLP is independent of the size of the domain in the lateral x and y directions (orthogonal to the movement of the structure-formation front). Hence, for the training data, one fixed lateral domain size is chosen. It is selected as small as possible, but such that all important characteristics are still captured.

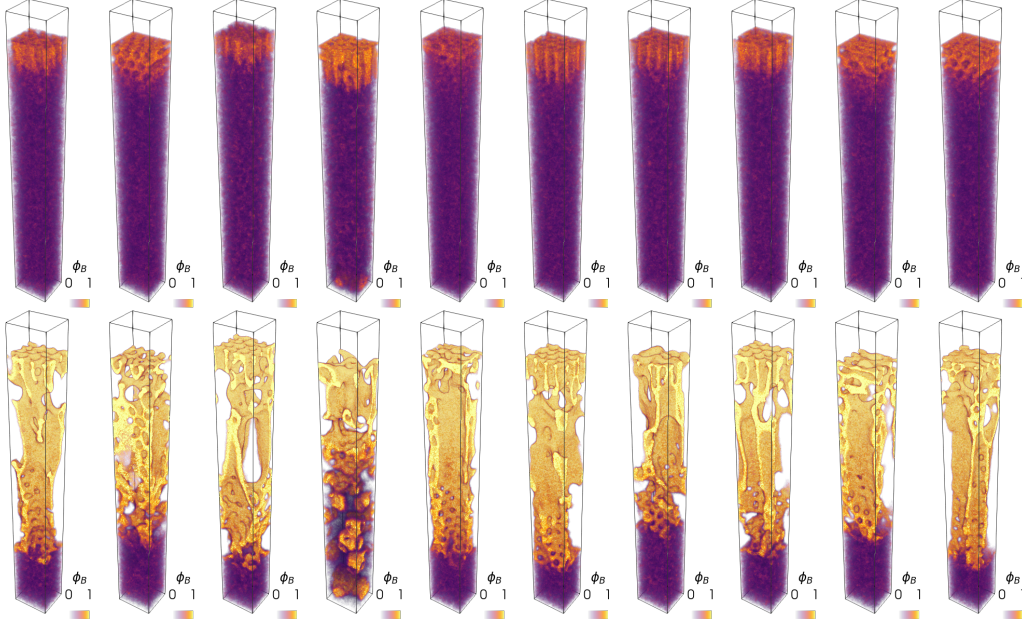


Figure 7: Example snapshots of 10 realizations (process parameters randomly varied between columns) of the training data, produced with the particle model. The top row shows the B -concentration fields, $\phi_B(\mathbf{r})$, after EISA, the bottom rows shows these after NIPS.

Our model is targeted to work on a large range of material and process parameters. For this reason, we generate 40 sets of parameter combinations that are close to the below base parameter combination but differ strongly enough that a number of different physically relevant morphologies emerge. To this end, we randomly generate the $\chi_{\alpha\beta}N_P$ interaction matrix from a Gaussian distribution, with the mean corresponding to the base-parameter combination and the standard deviation

$$\sqrt{\langle \chi N_P^2 \rangle - \langle \chi N_P \rangle^2} = \begin{pmatrix} & A & B & S & C & G & N \\ A & 0 & 2.5 & 2.5 & 0 & 15 & 5 \\ B & 2.5 & 0 & 0 & 1 & 25 & 25 \\ S & 2.5 & 0 & 0 & 0 & 10 & 10 \\ C & 0 & 1 & 0 & 0 & 10 & 10 \\ G & 15 & 25 & 10 & 10 & 0 & - \\ N & 5 & 25 & 10 & 10 & - & 0 \end{pmatrix} \quad (9)$$

In particular, we do not vary the interactions $\chi_{AC}N_P$ and $\chi_{BS}N_P$, because this would allow positive values, *i.e.*, repulsions of A to C , and B to S . In

experimental systems, the majority block, B typically has a preference for the volatile solvent S , while the minority block, A , typically is preferential towards the nonvolatile solvent C . All random parameter combinations should conform to this principle. Indeed, varying the above interaction, allowing positive values, resulted in many realizations for which the initial solution was immediately unstable to macrophase separation of polymer from the two solvents. This has no experimental relevance and, thus, also none to our simulation studies.

A particle simulation is run fully independently for each parameter combination, with $t_{\text{EISA}} = 5 \cdot 10^5 \text{MCS} \approx 16\tau_R$ and $t_{\text{NIPS}} = 7.5 \cdot 10^5 \text{MCS} \approx 24\tau_R$. The instantaneous concentration fields are saved every $t_{\text{save}} = 5000 \text{MCS} \approx 0.16\tau_R$. Every second of these saved morphologies then serves as an initial condition to a continuum simulation that is run for $t = 4.2\lambda_0^{-1}$, saving the concentration fields synchronously to the particle simulation, *i.e.*, every $t_{\text{save}} = 0.21\lambda_0^{-1}$.

The largest bottleneck of the training data generation is the label generation since it requires a full particle simulation as a reference solution to compare the continuum simulation with. Thus, this method of training data generation is efficient, since only a single particle simulation is run per parameter combination, while the repeated start of the computationally efficient continuum simulation generates a large amount of comparable time evolutions between the two models. Afterwards, a sample of training data consists of 21 labels and a set of descriptors for multiple layers, like visualized in Figure 5. Finally, generating the labels for the 40 different parameter combinations results in a total of $1.2 \cdot 10^6$ samples to train the MLP.

3.3. Post-processing

The goal of the post-processing step is to generate a reliable subdomain in which the higher-fidelity simulation must be applied. This subdomain is determined from the error predictions introduced in Section 3.2. From the whole range of time steps available, a subset of approximately the last quarter until the next prediction update is chosen. This accounts for the relevance of errors until the next update as well as effectively reducing time constant errors by only incorporating time steps where error peaks have already formed. In practice, the procedure combines several operations: an adaptive thresholding of the predicted error, identification of the dominant connected error region, Gaussian smoothing to reduce local noise, and aggregation of errors across the considered time horizon.

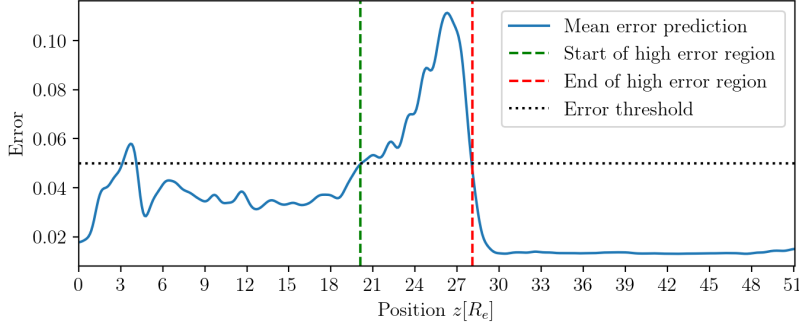


Figure 8: Mean error distribution for a single time step. The errors (Figure 9) are aggregated over a range of time steps approximately up to the point in time where the next prediction will be done. From this error distribution, one subdomain, where errors exceed the threshold the most, is identified as the part of the domain where the higher precision simulation is applied. The subdomain is marked by the green and red dashed lines, the threshold by a black dashed line.

An exemplary result is shown in Figure 8. Here, the ordinate represents the mean error, aggregated over multiple prediction steps, and the abscissa corresponds to the z -position of the layer. The black dashed line indicates an exemplary threshold at a value of 0.05. The resulting subdomain is bounded by green and red dashed lines, marking the beginning and end of the high-error region, respectively. Only one connected region is selected, namely the one with the highest error peak. In the illustrated case, two peaks are present in the mean error distribution. The higher peak is enclosed by the marking lines and corresponds to the structure-formation front, averaged over multiple time steps. In contrast, the second peak to the left is associated with the majorly solidified solution top layer. In the time evolution of Figure 5 it becomes visible that this peak appears immediately at the beginning of the continuum simulation and afterwards retains the same height. This means, there are some immediate short-time dynamics upon initialization with particle-simulation concentration fields, but no long-time changes occur, not requiring a long-time outsourcing of this region to the particle simulation. To the right of the structure-formation front, the mean error is close to zero, reflecting the homogeneous fluid.

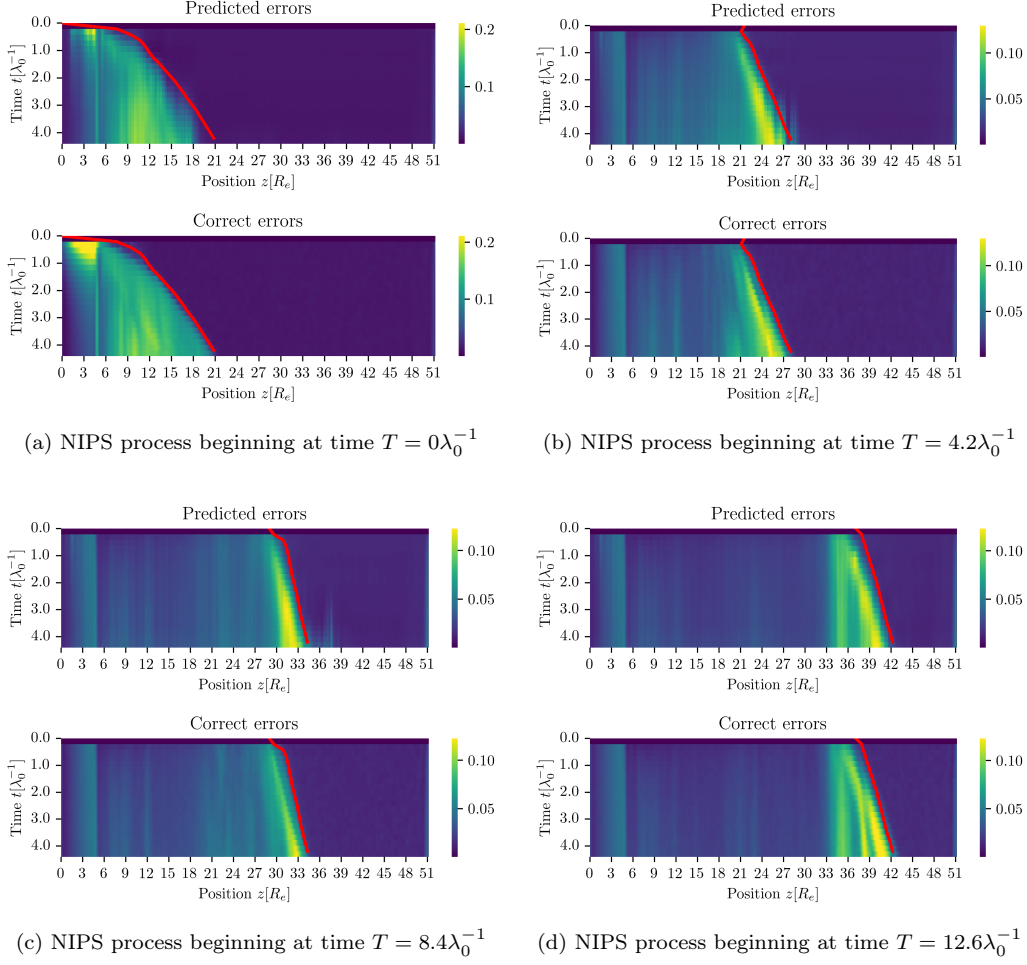


Figure 9: Heatmaps with predicted errors (upper images) in comparison to the correct errors (lower images). In each heatmap, the red line indicates the position of the structure-formation front, estimated by using the non-solvent density threshold $\phi_N^* = 0.0017$, at which instability against macrophase separation emerges (see Figure B.13 in Appendix B for the full Gibbs triangle). The data comes from an exemplary simulation which is not in the training set. For other simulations, predicted and correct errors exhibit similar dynamics. On the x -axis the layer coordinate shown, meaning this is the direction the structure-formation front is moving. The y -axis denotes the relative time step to the starting configuration, which is noted below the respective diagrams.

4. Results

4.1. Error prediction results

In Figure 9 a comparison between calculated and predicted errors is shown for a set of different time points during the NIPS simulation. The data stems from a simulation in the test set and was therefore not included in the MLP training. On the abscissa, the z -position (layer coordinate) is shown, while the ordinate denotes the time relative to the current simulation state as indicated in the respective captions. The red line marks an approximation of the structure-formation front, estimated by using the non-solvent density threshold $\phi_N = 0.0017$, at which instability against macrophase separation emerges on the Gibbs triangle as predicted by RPA (see Figure B.13). It shows that generally, the error between both simulations begins to grow directly where the structure formation starts.

The predicted errors show overall good qualitative agreement with the reference errors. In particular, the movement of the structure-formation front is clearly captured by the MLP, with error peaks shifting accordingly in space and time. The model reproduces the important features such as the location of maximum errors, the shape of error peaks, and the extent of high-error regions. Similarly, low-error regions are predicted with good accuracy.

Deviations appear mainly at later prediction times, which is expected, since forecasting further into the future generally becomes more challenging. At the beginning of the simulation, the very first predictions (Figure 9 a) also show notable discrepancies, likely because the simulation characteristics are not yet well established at this early stage. Nevertheless, even here the qualitative shape of the error distribution is reproduced. Finally, at the latest prediction time steps, some artifacts emerge in front of the structure formation region, located just ahead of the high-error zones (mainly visible in Figure 9 b and c). While these artifacts are not present in the ground truth, they remain confined to the later stages of the prediction. They might have their origin in noise of the input data and an unclear estimation of the structure-formation fronts' propagation speed.

4.2. Post-processing results

A more comprehensive comparison between predicted and reference post-processed error distributions is given in Figure 10. Results are shown for two test simulations not included in the training data. In each case, the upper diagrams display the predicted error distributions at several time steps, while

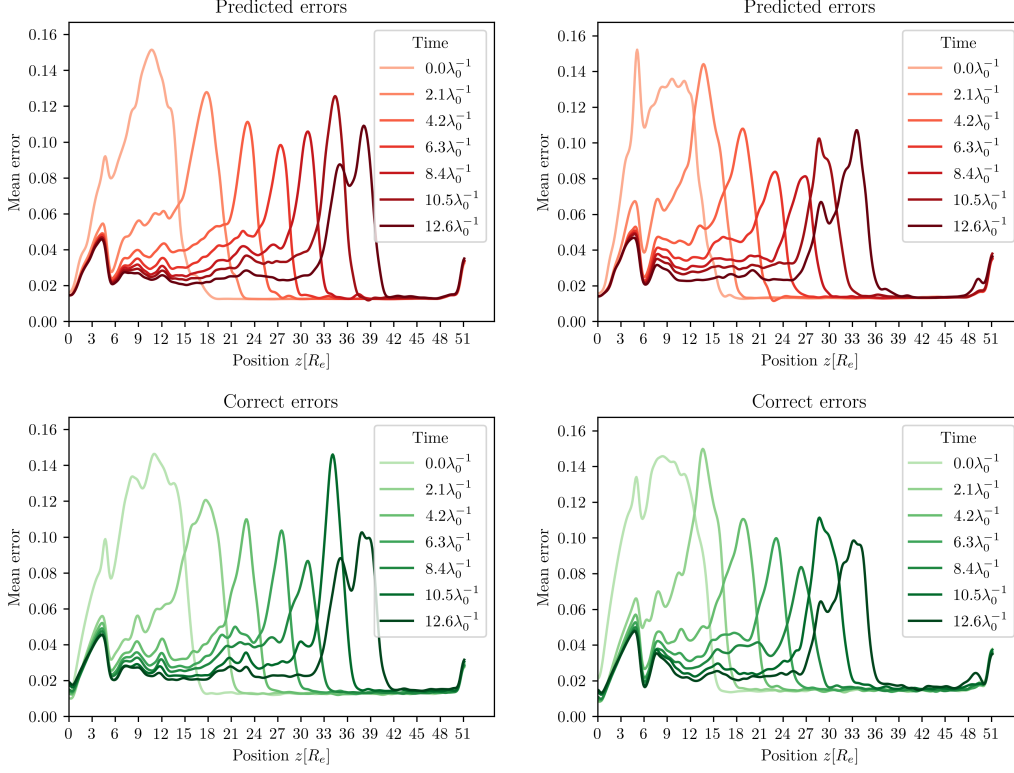


Figure 10: Mean error distributions for different time steps of two simulations. The respective upper diagram depicts the predictions (red colors), the lower diagram depicts the correct error values (green colors). Between the left and right diagrams the simulation parameterization is changed. Both simulations are excluded from the training data.

the lower diagrams show the corresponding ground-truth errors. Within each plot, the temporal evolution is illustrated by curves of varying color intensity. The comparison demonstrates that the predicted mean error distributions reproduce the key qualitative features of the reference: the location of the structure-formation front, the movement of the error peaks over time, and the characteristic shape of the distributions. Both simulations exhibit different speeds of front propagation as well as different magnitudes and temporal changes of the error. These differences are captured with good fidelity by the predictions, underlining the robustness of the post-processing approach.

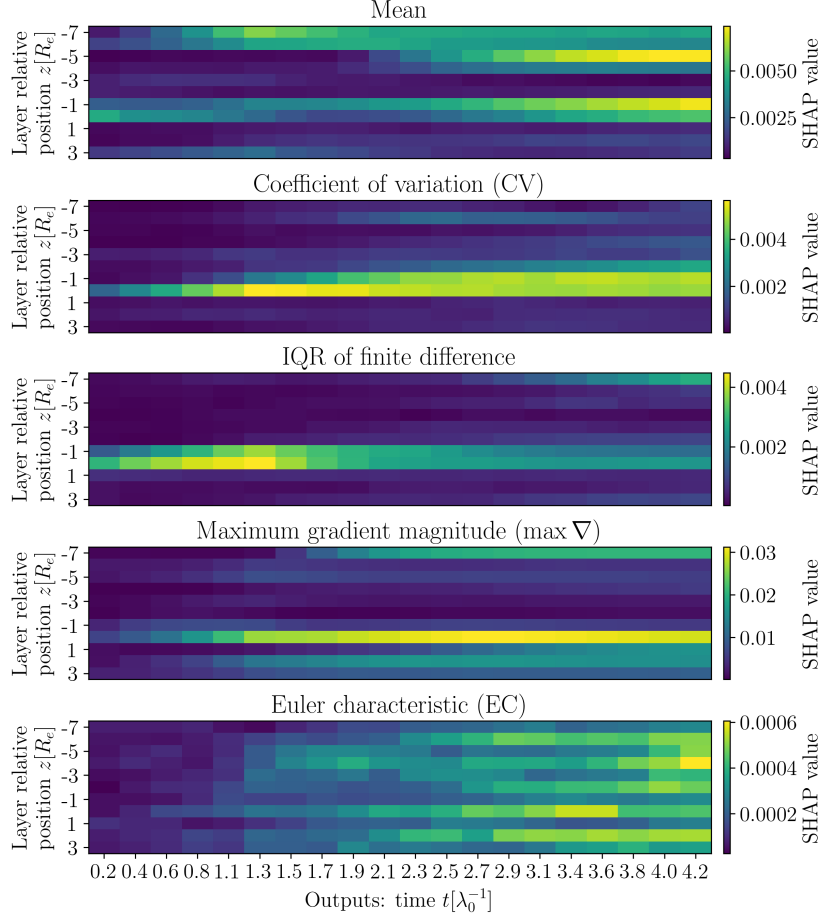


Figure 11: Results of the Shapley additive explanations (SHAP) analysis, sorted separately for each descriptor. Notably, the scale of the values varies significantly as can be seen at the color bars. All diagrams share the same x -axis, describing the different MLP outputs whereas the y -axis denotes the positions of the different input layers, relative to the current layer the prediction is made for.

4.3. Shapley additive explanations (SHAP) analysis

To better understand how the prediction MLP uses the input descriptors, we applied Shapley additive explanations (SHAP) to quantify the contribution of each input feature to each predicted output. The resulting SHAP matrix (Figure 11) contains the mean contribution values across all samples, sorted by descriptor. Each descriptor is shown with its own color scale, as absolute SHAP magnitudes vary by up to two orders of magnitude. The

y -axis represents the spatial offset of the input layers relative to the target layer in units of R_e , with $0R_e$ denoting the layer for which the divergence is predicted. The x -axis indicates the relative prediction time step. This representation allows us to identify both which descriptors are most influential overall and how their relevance changes across space and prediction horizon.

The SHAP analysis results show clear trends in how the MLP utilizes the different descriptors. As expected, the layer corresponding to the target position ($0R_e$) is generally the most influential, indicating that information from the current layer is most relevant for predicting its own divergence. For most descriptors, the SHAP values are small for short prediction horizons and increase for later output times, which is consistent with the observation that prediction errors also grow with the forecasting horizon.

Among the descriptors, the maximum gradient magnitude stands out as the most important by a substantial margin, with SHAP values roughly one order of magnitude larger than those of other descriptors. This descriptor is also the only one for which the MLP makes use of information from as far as three layers ahead of the target layer – likely reflecting its strong predictive value before the onset of structure formation. At later time steps, this descriptor is also used for the last available layer, which may relate to its relevance for estimating large errors far into the future. A similar pattern is observed for the interquartile range (IQR) of finite differences, which is most heavily used at the current and previous layer for early prediction times, suggesting that local structure formation directly informs near-term error growth.

The coefficient of variation (CV) shows a similar spatial usage pattern to the IQR, being most relevant at the current and previous layers, but it also retains moderate importance for longer-term predictions. The mean descriptor, in contrast, is primarily used for the most distant prediction horizons, indicating that large-scale average values play a role when detailed local information becomes less predictive.

Finally, the Euler characteristic shows consistently negligible SHAP values across all outputs and layers. Its contribution appears to be close to noise level, which may indicate that it is not a reliable predictor in this context or that any apparent signal is the result of overfitting in the descriptor selection process.

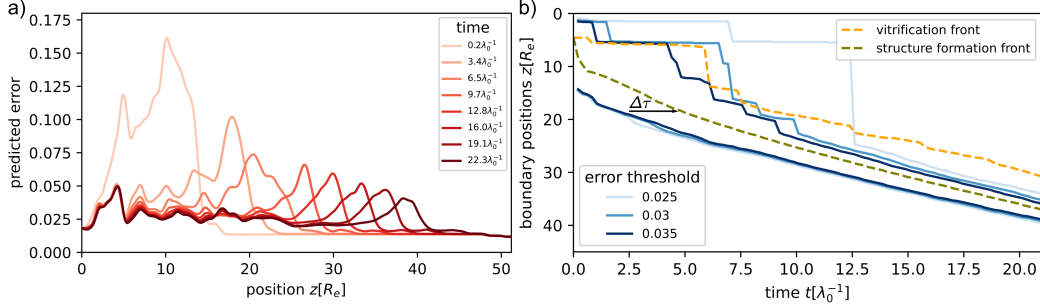


Figure 12: a) Error prediction of the continuum simulation compared to the particle simulations based on the pure-continuum simulation of Figure 4. The figure shows the mean error at different times during the NIPS process, plotted against the vertical position, z . b) Resulting sub-domain selection after postprocessing for different threshold values (solid) and for the region with the persistent peak in the cylindrical top-layer (around $z = 4R_e$, dashed).

4.4. Inference to continuum-simulation results

The training of the ML-model is based on particle simulations and the dynamics, from arbitrary transient morphologies as initial conditions, within the continuum model. Now, let us apply the ML-based error prediction to the continuum model simulation presented above. Here we choose the one with weaker solvent attraction, $\chi_{SC}N_P = -28$, compared to the base parameters, Figure 4, which resulted in a continuous membrane morphology in both simulation models.

Figure 12 a) shows the mean error prediction for the error after time $\Delta\tau = 2.4\lambda_0^{-1}$, for the continuum simulation at different times during the NIPS process. Generally, the error prediction for the full continuum simulation adopts the same shape as the prediction for the test data, Figure 10, also in the case of a larger time horizon compared to the training data. The largest error is predicted to occur at early times at the top of the solution film. The error peak then travels through the film, but a minor peak remains at the top of the film.

Figure 12 b) shows the position of the proposed sub-domains for three different error thresholds, based on the error prediction. As a comparison, the figure shows the approximation of the vitrification and structure-formation front, introduced in subsection 2.4. As indicated, the ML-proposed boundaries bracket the structure-formation front and lead by exactly the distance that the front travels during the outsource time $\Delta\tau$. This means that the

ML-model accurately predicts the speed of the structure-formation front. Furthermore, this simulation exceeds the time horizon of the training and shows that the error prediction generalizes to larger time steps, which are not included in the training data.

On the other end of the subdomain, the vitrification front is encapsulated at the beginning of the simulation, whereas at the later times it is excluded. Hence, the ML-model predicts the later stages of structure formation, just before vitrification occurs with sufficient precision in the UDM.

5. Discussion

This study introduced a hybrid simulation scenario of a membrane fabrication process combining two complementary models: a particle approach providing high fidelity at the cost of computational speed, and a continuum-based approach offering higher efficiency at reduced accuracy. The investigated simulation scenario is characterized by a structure-formation front propagating along one coordinate direction, which serves as a representative test case for adaptive multi-model coupling.

To enable dynamic switching between these models, an error prediction mechanism based on machine learning was developed. As shown in Figure 5, the proposed decision model leverages two-dimensional descriptors and a layer-wise prediction strategy to estimate local model errors. These predictions form the basis for adaptive model selection, ensuring that computational resources are concentrated where accuracy is most critical while maintaining overall efficiency.

The results of the prototype implementation confirm that the proposed decision model enables adaptive coupling of two different simulation models in practice. As illustrated in Figure 5, the architecture bases decisions on predicted errors and thereby achieves efficient resource allocation while maintaining fidelity in regions of interest.

A necessary ability of the model is to generalize to a wide range of process- and material-parameter combinations of NIPS, which the proposed architecture satisfies. It performs independently of the domain size in all three coordinate directions, extrapolates to later time steps not included in training, and remains robust under varying simulation parameters within the scope of the present study. These properties distinguish the approach from more narrowly parameterized surrogate models.

The error predicting MLP and the layer-wise prediction architecture, however, are closely tied to the characteristics of the underlying system. In our case, the presence of a structure-formation front with localized dynamics motivated the choice of simplification to a 1D error prediction and localization strategy. Applications to other systems will require adapting these components to the relevant physical features.

Overall, the results demonstrate that machine-learned error prediction provides a viable route toward scalable, adaptive multi-fidelity coupling. Future work will integrate the present prototype in fully coupled simulations and systematic benchmarks, with the aim of establishing general guidelines for applying the framework to other classes of multiscale problems.

6. Acknowledgements

We gratefully acknowledge the support of the Helmholtz-Gemeinschaft Deutscher Forschungszentren (HGF). The authors gratefully acknowledge the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS Supercomputer JUWELS at Jülich Supercomputing Centre (JSC).

7. Funding

Financial support was provided by the Bundesministerium für Forschung, Technologie und Raumfahrt (BMFTR) within the project 16ME0658K MExMeMo.

8. Declaration of competing interests

The authors declare no competing interests.

9. Data availability

The training data for the machine learning model, the model itself, as well as workflows and input files for the two simulation schemes and training data generation are provided on Zenodo: <https://doi.org/10.5281/zenodo.17396869>.

10. Code availability

The source code for reproducing the results in this publication is provided on Zenodo: <https://doi.org/10.5281/zenodo.17396869>.

11. Reproducibility statement

With the provided code and data, the results are reproducible. Figure 6 and Figure 8 use data from the simulation with 0-based index 20, Figure 9 from simulation 19, and Figure 10 from simulations 19 (left) and 18 (right).

12. Author contribution: CRediT

Matthias Busch: Conceptualization, Methodology, Software, Investigation, Formal Analysis, Writing - Original Draft, Visualization. **Gregor Häfner:** Conceptualization, Methodology, Software, Investigation, Formal Analysis, Writing - Original Draft, Visualization. **Jiayu Xie:** Methodology, Writing - Review & Editing. **Marius Tacke:** Writing - Review & Editing. **Marcus Müller:** Funding Acquisition, Supervision, Conceptualization, Writing - Review & Editing. **Christian J. Cyron:** Funding Acquisition, Writing - Review & Editing. **Roland C. Aydin:** Supervision, Conceptualization, Writing - Review & Editing.

13. Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used ChatGPT in order to review the manuscript for errors and possible improvements. This includes rephrasing parts of the manuscript. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Appendix A. Simulation techniques

Appendix A.1. Continuum Model

The continuum model employs a free-energy functional approach based on the Uneyama-Doi free-energy functional that generalizes the Ohta-Kawasaki theory for diblock copolymers [46] to the strong segregation regime and

blends with homopolymers and solvents. The system is described by normalized local concentration fields $\phi_\alpha(\mathbf{r})$ of component α at position $\mathbf{r}$, where α collectively denotes molecular species and their blocks. The total free energy functional reads [35, 36]

$$\begin{aligned} \frac{\mathcal{F}[\{\phi_\alpha(\mathbf{r})\}]}{\sqrt{N}k_B T} = & \frac{1}{R_e^3} \int d\mathbf{r} \left[P(\mathbf{r}) N_0 \left(\sum_\alpha \phi_\alpha - 1 \right) + \frac{1}{2} \sum_{\alpha \neq \beta} \chi_{\alpha\beta} N_0 \phi_\alpha(\mathbf{r}) \phi_\beta(\mathbf{r}) \right. \\ & + \sum_\alpha A_{\alpha\beta} \sqrt{\phi_\alpha(\mathbf{r})} \int d\mathbf{r}' \mathcal{G}(\mathbf{r} - \mathbf{r}') \sqrt{\phi_\beta(\mathbf{r}')} \\ & \left. + \sum_\alpha C_{\alpha\beta} \phi_\alpha(\mathbf{r}) \ln \phi_\alpha(\mathbf{r}) + \sum_\alpha \frac{R_e^2}{24} \frac{|\nabla \phi_\alpha(\mathbf{r})|^2}{\phi_\alpha(\mathbf{r})} \right], \end{aligned} \quad (\text{A.1})$$

where $P(\mathbf{r})$ is a pressure-like Lagrange multiplier enforcing incompressibility, and $A_{\alpha\beta}$, $C_{\alpha\beta}$ are architecture-dependent coefficients. The second term describes enthalpic interactions via Flory-Huggins parameters $\chi_{\alpha\beta}$, the third term accounts for covalent bonding through a long-range interaction with kernel $\mathcal{G}(\mathbf{r})$, satisfying $(-\nabla^2 + \xi_{\text{cut}}^{-2})\mathcal{G}(\mathbf{r}) = \delta(\mathbf{r})$, the fourth term captures the entropy of mixing, and the fifth term imposes interfacial width via a gradient penalty.

For the AB diblock copolymer with block ratio $f_A = 1 - f_B$, the architecture coefficients are

$$A = \frac{9N_0^2}{R_e^2 f_A^2 f_B^2} \begin{pmatrix} f_B^2 & -f_A f_B \\ -f_A f_B & f_A^2 \end{pmatrix}, \quad C = \frac{N_0}{N_p} \begin{pmatrix} \frac{\tilde{s}(f_A)}{f_B} & \frac{-1}{2\sqrt{f_A f_B}} \\ \frac{-1}{2\sqrt{f_A f_B}} & \frac{\tilde{s}(f_B)}{f_A} \end{pmatrix}, \quad (\text{A.2})$$

with $\tilde{s}(f) = [s(f) - f]/[4f(1 - f)]$. Here, the function $s(f) = 1.572 - 2.702f(1 - f)$ is a fitting function that ensures consistency with the order-disorder transition predicted by RPA [46]. For solvents and homopolymers, $A_\alpha = 0$ and $C_\alpha = N_0/N_\alpha$.

The temporal evolution is governed by model-B dynamics [34] combining diffusive fluxes with conversion source/sink terms during the SNIPS process. This results in Equation 1, where $\mu_\alpha(\mathbf{r}) = \delta\mathcal{F}/\delta\phi_\alpha(\mathbf{r})$ is the chemical potential and $s_\alpha(\mathbf{r}, t)$ describes molecular conversions.

In the time evolution equation, the mobility, $\lambda_\alpha(\mathbf{r}, t)/\lambda_0$, follows the same functional form as the inverse segmental friction in the particle model, $m_{A/B}$, accounts for vitrification. Weak thermal noise is incorporated through Gaussian random fields satisfying the fluctuation-dissipation relation, as described

in Ref. [29]. In the here-presented simulations, thermal fluctuations have strength $\sqrt{\mathcal{N}} = 10^5$, only during the EISA process. Hence, fluctuations are essentially not present, but weak lateral fluctuations are required in the first process step, for their amplitude to grow after the onset of microphase separation. Conversions are performed both to model solvent evaporation, as well as the solvent-nonsolvent exchange. In both cases, these are performed in a conversion zone, which follows the interface position, z_I , at a constant distance of $\Delta z_{cz} = 0.6R_e$ during the EISA process and is fixed during the NIPS process. Hence the following conversion occurs exclusively for $z < z_I - \Delta z_{cz}$. For the EISA process, solvent is converted to gas, $S \rightarrow G$, specifically $s_G = -s_S = r\phi_S$. For NIPS, both the S and C solvent are exchanged for nonsolvent in the solution film, and therefore both are converted to nonsolvent at $z = 0$. Here, $s_C = -r\phi_C$, $s_S = -r\phi_S$, and $s_N = r(\phi_S + \phi_C)$. In all cases the conversion occurs at a high rate, $r = 10^3\lambda_0$, practically instantly converting all material.

These continuum dynamics are flexibly and efficiently implemented within the UDM-software in CUDA/C for GPU acceleration. The software is available open-source *via* www.gitlab.com/g.ibbeken/udm.

Appendix A.2. Particle-based Simulations

The particle simulations employ a soft, coarse-grained model [30] that represents several monomer repeat units by a single particle. Within the soft, coarse-grained model the Hamiltonian is split into strong bonded (b) and weak non-bonded (nb) interactions, $\mathcal{H} = \mathcal{H}_b + \mathcal{H}_{nb}$.

The strong bonded interactions are taken to be harmonic springs

$$\frac{\mathcal{H}_b}{k_B T} = \sum_m \sum_b \frac{3(N_0 - 1)}{2R_e^2} (\mathbf{r}_{m,b} - \mathbf{r}_{m,b+1})^2, \quad (\text{A.3})$$

where m indexes molecules and b indexes bonds within each molecule. $\mathbf{r}_{m,b}$, $\mathbf{r}_{m,b+1}$ refer to the positions of the bonded particles. Further, k_B is the Boltzmann constant, T the temperature, N_0 refers to the chain-contour discretization of the polymer, and R_e to its root mean-squared end-to-end distance.

The weak non-bonded interactions are expressed in terms of the normal-

ized densities $\phi_\alpha(\mathbf{r})$ of component α at position $\mathbf{r}$, *i.e.*,

$$\frac{\mathcal{H}_{\text{nb}}}{\sqrt{\mathcal{N}}k_{\text{B}}T} = \int \frac{d\mathbf{r}}{R_e^3} \left(\frac{\kappa_0 N_0}{2} \left[\sum_{\alpha} \phi_{\alpha}(\mathbf{r}) - 1 \right]^2 + \frac{1}{2} \sum_{\alpha \neq \beta} \chi_{\alpha\beta} N_0 \phi_{\alpha}(\mathbf{r}) \phi_{\beta}(\mathbf{r}) \right), \quad (\text{A.4})$$

where κ_0 characterizes the inverse compressibility of the system, $\chi_{\alpha\beta}$ are the Flory-Huggins parameters describing the binary repulsion between the species α and β . The first term enforces near incompressibility and the second one accounts for the binary interactions of different components. $\sqrt{\mathcal{N}} = nR_e^3/(VN_0)$ denotes the invariant degree of polymerization and sets the strength of thermal fluctuations. n denotes the total number of beads in the cubic simulation cell of volume V with periodic boundary conditions. The densities are calculated on a cubic collocation grid with linear spacing $\Delta x = \Delta y = \Delta z$.

The system comprises an AB diblock copolymer and multiple solvent types to model the SNIPS process: a volatile solvent S (THF), a nonvolatile solvent C (DMF), a nonsolvent N (water), and gas molecules G (air). The interactions of the base parameter combination are given in the main text.

We employ the SCMF algorithm [30, 32] that temporarily replaces the weak, non-bonded interactions by external fields and thereby exploits the different strengths of strong bonded and weak but computationally costly non-bonded interactions. Particle positions are updated by the smart-Monte-Carlo algorithm, using the strong bonded forces to bias the trial displacement. The time it takes a copolymer to diffuse its own mean end-to-end distance, R_e , in a disordered system serves as the time unit τ_R .

To account for vitrification during the membrane formation process, polymer segments A and B are slowed down depending on the local concentration fields via the inverse segment friction $m(\{\phi_{\alpha}\})$. The inverse polymer segment friction varies with local concentration according to

$$m(\{\phi_{\alpha}\}) = \frac{1}{2} \left[1 + \tanh \left(\frac{1 - \sum_{\alpha} a_{\alpha} \phi_{\alpha}}{3} \right) \right] \quad (\text{A.5})$$

with coefficients a_{α} the position, ϕ_P^* and width of the vitrification threshold, *i.e.* at which polymer concentration the polymer dynamics become arrested.

Specifically, $a_A = a_B = 17$, $a_S = a_C = -40$, $a_G = a_N = 0$ were chosen, resulting in $\phi_P^* \approx 0.72$. The solvent mobilities were fixed to the value $m = 0.88$.

The SNIPS process is modeled by solvent exchange. During EISA, volatile solvent molecules S reaching a conversion zone at the film surface are converted to gas molecules G, resulting in film surface retraction. The conversion zone tracks the film surface position at a fixed offset. To initiate NIPS, all gas molecules G are converted to nonsolvent N, and subsequently both solvents S and C are converted to N in the conversion zone. The positioning of the conversion zone occurs in the same manner as in the continuum model, described above.

We use the highly parallel and graphics processing unit (GPU)-accelerated software SOMA[32], available open-source via www.gitlab.com/InnocentBug/SOMA.

Appendix B. Gibbs triangle

The Gibbs triangle corresponding to the random parameter combination of the testing data in subsection 4.1 is given in Figure B.13. From this and more resolved nonsolvent concentration ϕ_N , the threshold concentration for macrophase separation to occur was determined to $\phi_N^* = 0.0017$.

References

- [1] M. Müller, V. Abetz, Nonequilibrium Processes in Polymer Membrane Formation: Theory and Experiment, *Chem. Rev.* 121 (22) (2021) 14189–14231. doi:10.1021/acs.chemrev.1c00029.
- [2] W. E, B. Engquist, The heterogeneous multiscale methods, *Communications in Mathematical Sciences* 1 (1) (2003) 87–132. doi:10.4310/CMS.2003.v1.n1.a8.
- [3] M. Müller, K. Ch. Daoulas, Speeding Up Intrinsically Slow Collective Processes in Particle Simulations by Concurrent Coupling to a Continuum Description, *Phys. Rev. Lett.* 107 (22) (2011) 227801. doi:10.1103/PhysRevLett.107.227801.
- [4] E. B. Tadmor, M. Ortiz, R. Phillips, Quasicontinuum analysis of defects in solids, *Philosophical Magazine A* 73 (6) (1996) 1529–1563. doi:10.1080/01418619608243000.

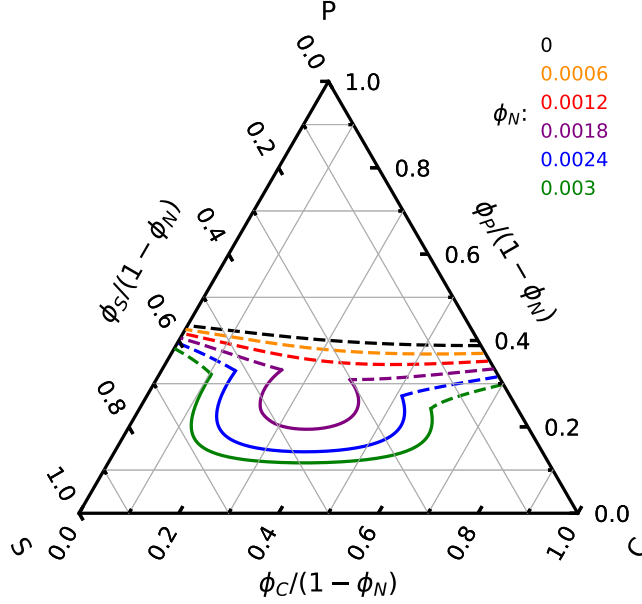


Figure B.13: Spinodal curves of membrane casting solutions at different nonsolvent concentrations on the Gibbs triangle. The system parameters are identical to those used in Figure 9. Dashed and solid lines indicate instabilities of the spatially homogeneous state against composition fluctuations with finite and infinite length scales, respectively.

- [5] T. Belytschko, S. Xiao, Coupling methods for continuum model with molecular model, *International Journal for Multiscale Computational Engineering - INT J MULTISCALE COMPUT ENG* 1 (2003) 115–126. doi:10.1615/IntJMultCompEng.v1.i1.100.
- [6] B. Peherstorfer, K. Willcox, M. Gunzburger, Survey of multifidelity methods in uncertainty propagation, inference, and optimization, *SIAM Review* 60 (3) (2018) 550–591. doi:10.1137/16M1082469.
- [7] K. Duraisamy, G. Iaccarino, H. Xiao, Turbulence modeling in the age of data, *Annual Review of Fluid Mechanics* 51 (Volume 51, 2019) (2019) 357–377. doi:10.1146/annurev-fluid-010518-040547.
- [8] M. Raissi, G. Karniadakis, Deep multi-fidelity gaussian processes (04 2016). doi:10.48550/arXiv.1604.07484.
- [9] P. Perdikaris, M. Raissi, A. Damianou, N. D. Lawrence, G. E. Karniadakis, Nonlinear information fusion algorithms for data-efficient multi-

- fidelity modelling, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 473 (2198) (2017) 20160751. doi:10.1098/rspa.2016.0751.
- [10] E. J. Parish, K. T. Carlberg, Time-series machine-learning error models for approximate solutions to parameterized dynamical systems, *Computer Methods in Applied Mechanics and Engineering* 365 (2020) 112990. doi:<https://doi.org/10.1016/j.cma.2020.112990>.
 - [11] B. Sanderse, P. Stinis, R. Maulik, S. E. Ahmed, Scientific machine learning for closure models in multiscale problems: a review (2024). [arXiv:2403.02913](https://arxiv.org/abs/2403.02913).
 - [12] P. C. H. Nguyen, J. B. Choi, H. S. Udaykumar, S. Baek, Challenges and opportunities for machine learning in multiscale computational modeling (2023). [arXiv:2303.12261](https://arxiv.org/abs/2303.12261).
 - [13] H. Bhatia, T. S. Carpenter, H. I. Ingólfsson, G. Dharuman, P. Karande, S. Liu, T. Oppelstrup, C. Neale, F. C. Lightstone, B. Van Essen, et al., Machine-learning-based dynamic-importance sampling for adaptive multiscale simulations, *Nature Machine Intelligence* 3 (5) (04 2021). doi:10.1038/s42256-021-00327-w.
 - [14] H. I. Ingólfsson, C. Neale, T. S. Carpenter, R. Shrestha, C. A. López, T. H. Tran, T. Oppelstrup, H. Bhatia, L. G. Stanton, X. Zhang, S. Sundram, F. D. Natale, A. Agarwal, G. Dharuman, S. I. L. K. Schumacher, T. Turbyville, G. Gulten, Q. N. Van, D. Goswami, F. Jean-Francois, C. Agamasu, D. Chen, J. J. Hettige, T. Travers, S. Sarkar, M. P. Surh, Y. Yang, A. Moody, S. Liu, B. C. V. Essen, A. F. Voter, A. Ramanathan, N. W. Hengartner, D. K. Simanshu, A. G. Stephen, P.-T. Bremer, S. Gnanakaran, J. N. Glosli, F. C. Lightstone, F. McCormick, D. V. Nissley, F. H. Streitz, Machine learning-driven multiscale modeling reveals lipid-dependent dynamics of ras signaling proteins, *Proceedings of the National Academy of Sciences* 119 (1) (2022) e2113297119. doi:10.1073/pnas.2113297119.
 - [15] K.-V. Peinemann, V. Abetz, P. F. W. Simon, Asymmetric superstructure formed in a block copolymer via phase separation, *Nature Materials* 6 (12) (2007) 992–996. doi:10.1038/nmat2038.

- [16] S. Rangou, K. Buhr, V. Filiz, J. I. Clodt, B. Lademann, J. Hahn, A. Jung, V. Abetz, Self-organized isoporous membranes with tailored pore sizes, *Journal of Membrane Science* 451 (2014) 266–275. doi:10.1016/j.memsci.2013.10.015.
- [17] V. Abetz, Isoporous Block Copolymer Membranes, *Macromolecular Rapid Communications* 36 (1) (2015) 10–22. doi:10.1002/marc.201400556.
- [18] M. Radjabian, V. Abetz, Tailored Pore Sizes in Integral Asymmetric Membranes Formed by Blends of Block Copolymers, *Advanced Materials* 27 (2) (2015) 352–355. doi:10.1002/adma.201404309.
- [19] L. Guo, Y. Wang, M. Steinhart, Porous block copolymer separation membranes for 21st century sanitation and hygiene, *Chemical Society Reviews* 50 (11) (2021) 6333–6348. doi:10.1039/D0CS00500B.
- [20] S. P. Paradiso, K. T. Delaney, C. J. García-Cervera, H. D. Cenicerros, G. H. Fredrickson, Block Copolymer Self Assembly during Rapid Solvent Evaporation: Insights into Cylinder Growth and Stability, *ACS Macro Letters* 3 (1) (2014) 16–20. doi:10.1021/mz400572r.
- [21] A. V. Berezkin, C. M. Papadakis, I. I. Potemkin, Vertical Domain Orientation in Cylinder-Forming Diblock Copolymer Films upon Solvent Vapor Annealing, *Macromolecules* 49 (1) (2016) 415–424. doi:10.1021/acs.macromol.5b01771.
- [22] J. Hao, Z. Wang, Z. Wang, Y. Yin, R. Jiang, B. Li, Q. Wang, Self-Assembly in Block Copolymer Thin Films upon Solvent Evaporation: A Simulation Study, *Macromolecules* 50 (11) (2017) 4384–4396. doi:10.1021/acs.macromol.7b00200.
- [23] O. Dreyer, G. Ibbeken, L. Schneider, N. Blagojevic, M. Radjabian, V. Abetz, M. Müller, Simulation of Solvent Evaporation from a Diblock Copolymer Film: Orientation of the Cylindrical Mesophase, *Macromolecules* 55 (17) (2022) 7564–7582. doi:10.1021/acs.macromol.2c00612.
- [24] D. J. Grzetic, A. J. Cooper, K. T. Delaney, G. H. Fredrickson, Modeling Microstructure Formation in Block Copolymer Membranes Using Dy-

- namical Self-Consistent Field Theory, *ACS Macro Letters* 12 (1) (2023) 8–13. doi:10.1021/acsmacrolett.2c00611.
- [25] A. J. Cooper, D. J. Grzetic, K. T. Delaney, G. H. Fredrickson, Investigating microstructure evolution in block copolymer membranes, *The Journal of Chemical Physics* 160 (7) (2024) 074903. doi:10.1063/5.0188196.
 - [26] N. Blagojevic, M. Müller, Simulation of Membrane Fabrication via Solvent Evaporation and Nonsolvent-Induced Phase Separation, *ACS Appl. Mater. Interfaces* 15 (50) (2023) 57913–57927. doi:10.1021/acsmi.3c03126.
 - [27] N. Blagojevic, S. Das, J. Xie, O. Dreyer, M. Radjabian, M. Held, V. Abetz, M. Müller, Toward Predicting the Formation of Integral-Asymmetric, Isoporous Diblock Copolymer Membranes, *Adv. Mater.* 36 (40) (2024) 2404560. doi:10.1002/adma.202404560.
 - [28] G. Häfner, M. Müller, Reaction-driven assembly: Controlling changes in membrane topology by reaction cycles, *Soft Matter* 19 (38) (2023) 7281–7292. doi:10.1039/D3SM00876B.
 - [29] G. Häfner, M. Müller, Reaction-Driven Diffusiophoresis of Liquid Condensates: Potential Mechanisms for Intracellular Organization, *ACS Nano* (Jun. 2024). doi:10.1021/acsnano.3c12842.
 - [30] K. C. Daoulas, M. Müller, Single chain in mean field simulations: Quasi-instantaneous field approximation and quantitative comparison with Monte Carlo simulations, *The Journal of Chemical Physics* 125 (18) (2006) 184904. doi:10.1063/1.2364506.
 - [31] M. Müller, Studying Amphiphilic Self-assembly with Soft Coarse-Grained Models, *Journal of Statistical Physics* 145 (4) (2011) 967–1016. doi:10.1007/s10955-011-0302-z.
 - [32] L. Schneider, M. Müller, Multi-architecture Monte-Carlo (MC) simulation of soft coarse-grained polymeric materials: Soft coarse grained Monte-Carlo Acceleration (SOMA), *Computer Physics Communications* 235 (2019) 463–476. doi:10.1016/j.cpc.2018.08.011.

- [33] M. Müller, K. Ch. Daoulas, Single-chain dynamics in a homogeneous melt and a lamellar microphase: A comparison between Smart Monte Carlo dynamics, slithering-snake dynamics, and slip-link dynamics, *J. Chem. Phys.* 129 (16) (2008) 164906. doi:10.1063/1.2997345.
- [34] P. C. Hohenberg, B. I. Halperin, Theory of dynamic critical phenomena, *Reviews of Modern Physics* 49 (3) (1977) 435–479. doi:10.1103/RevModPhys.49.435.
- [35] T. Uneyama, M. Doi, Density Functional Theory for Block Copolymer Melts and Blends, *Macromolecules* 38 (1) (2005) 196–205. doi:10.1021/ma049385m.
- [36] T. Uneyama, M. Doi, Calculation of the Micellar Structure of Polymer Surfactant on the Basis of the Density Functional Theory, *Macromolecules* 38 (13) (2005) 5817–5825. doi:10.1021/ma0507746.
- [37] M. W. Matsen, M. Schick, Stable and unstable phases of a diblock copolymer melt, *Physical Review Letters* 72 (16) (1994) 2660–2663. doi:10.1103/PhysRevLett.72.2660.
- [38] P.-G. De Gennes, *Scaling concepts in polymer physics*, Cornell university press, 1979.
- [39] L. Leibler, Theory of microphase separation in block copolymers, *Macromolecules* 13 (6) (1980) 1602–1617.
- [40] K. M. Hong, J. Noolandi, Theory of phase equilibria in systems containing block copolymers, *Macromolecules* 16 (7) (1983) 1083–1093.
- [41] M. D. Whitmore, J. Noolandi, Theory of phase equilibria in block copolymer-homopolymer blends, *Macromolecules* 18 (12) (1985) 2486–2497.
- [42] J. Xie, A.-C. Shi, Phase behavior of triblock copolymer and homopolymer blends: Effect of copolymer topology, *Physical Review Materials* 8 (1) (2024) 015601.
- [43] J. Xie, A.-C. Shi, Topological effects on the lifshitz point of block copolymer/homopolymer blends, *Physical Review Materials* 9 (1) (2025) 015601.

- [44] A. I. Awad, M. Hassaballah (Eds.), Image Feature Detectors and Descriptors, Studies in Computational Intelligence, Springer, Cham, Switzerland, 2016. doi:10.1007/978-3-319-28854-3.
- [45] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.
- [46] T. Ohta, K. Kawasaki, Equilibrium morphology of block copolymer melts, Macromolecules 19 (10) (1986) 2621–2632. doi:10.1021/ma00164a028.