

UvA-DARE (Digital Academic Repository)

Adversarial reverse mapping of equilibrated condensed-phase molecular structures

Stieffenhofer, M.; Wand, M.; Bereau, T.

DOI

[10.1088/2632-2153/abb6d4](https://doi.org/10.1088/2632-2153/abb6d4)

Publication date

2020

Document Version

Final published version

Published in

Machine Learning: Science and Technology

License

CC BY

[Link to publication](#)

Citation for published version (APA):

Stieffenhofer, M., Wand, M., & Bereau, T. (2020). Adversarial reverse mapping of equilibrated condensed-phase molecular structures. *Machine Learning: Science and Technology*, 1(4), [045014]. <https://doi.org/10.1088/2632-2153/abb6d4>

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (<https://dare.uva.nl>)



PAPER

OPEN ACCESS

RECEIVED

23 June 2020

REVISED

12 August 2020

ACCEPTED FOR PUBLICATION

9 September 2020

PUBLISHED

15 October 2020

Original Content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](#).

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.



Adversarial reverse mapping of equilibrated condensed-phase molecular structures

Marc Stieffenhofer¹, Michael Wand² and Tristan Bereau^{1,3} ¹ Max Planck Institute for Polymer Research, 55128 Mainz, Germany² Institute of Informatics, Johannes Gutenberg University, 55099 Mainz, Germany³ Van't Hoff Institute for Molecular Sciences and Informatics Institute, University of Amsterdam, Amsterdam 1098 XH, The NetherlandsE-mail: t.bereau@uva.nl**Keywords:** backmapping, multiscale modeling, generative adversarial networks, molecular dynamics, condensed phaseSupplementary material for this article is available [online](#)

Abstract

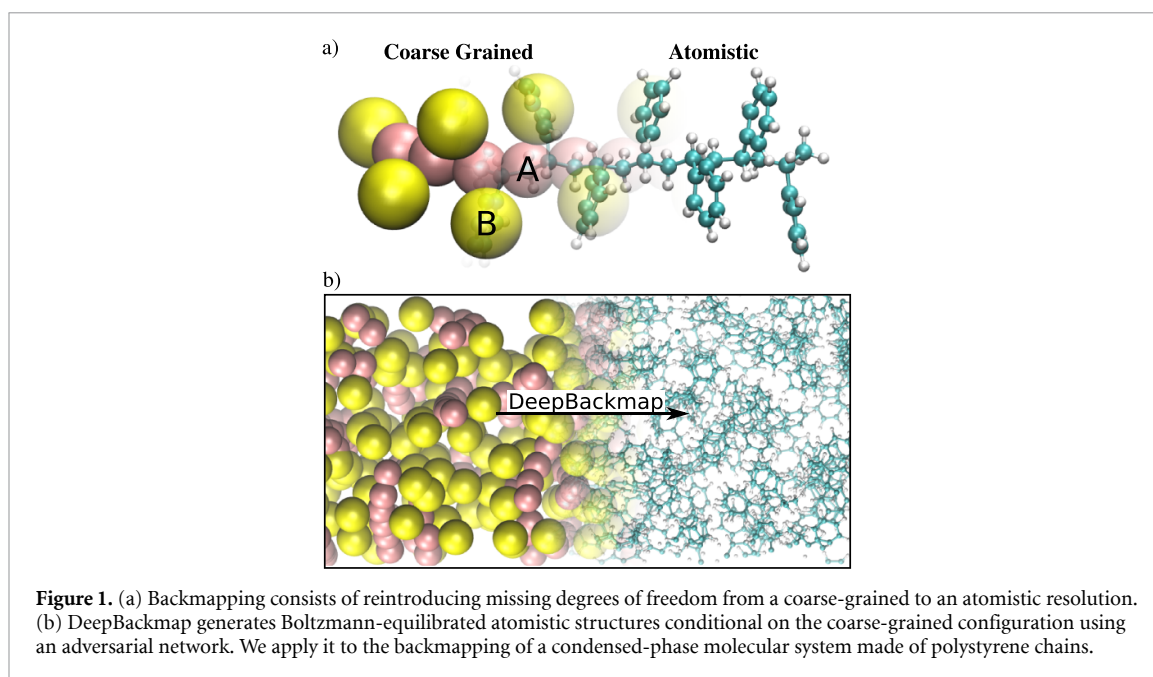
A tight and consistent link between resolutions is crucial to further expand the impact of multiscale modeling for complex materials. We herein tackle the generation of condensed molecular structures as a refinement—backmapping—of a coarse-grained (CG) structure. Traditional schemes start from a rough coarse-to-fine mapping and perform further energy minimization and molecular dynamics simulations to equilibrate the system. In this study we introduce DeepBackmap: A deep neural network based approach to directly predict equilibrated molecular structures for condensed-phase systems. We use generative adversarial networks to learn the Boltzmann distribution from training data and realize reverse mapping by using the CG structure as a conditional input. We apply our method to a challenging condensed-phase polymeric system. We observe that the model trained in a melt has remarkable transferability to the crystalline phase. The combination of data-driven and physics-based aspects of our architecture help reach temperature transferability with only limited training data.

1. Introduction

Computational modeling of soft-matter systems inherently requires the consideration of a wide range of time and length scales, where microscopic interactions can impact meso- to macroscopic changes [1]. Setting aside quantum mechanics, even force-field-based molecular dynamics (MD) quickly reaches its limits when probing long relaxation times. Circumventing such limitations remains an area of active research, motivated in part by the promises of computational soft materials discovery [2]. Various strategies aim at breaking the natural limitations of MD, from enhanced-sampling techniques [3] to dedicated hardware [4] to hierarchical multiscale modeling [1, 5, 6].

Multiscale modeling relies on several levels of resolution, striving to make best use of each level. At the lower end, a coarse-grained (CG) resolution will map groups of atoms to single interaction sites or beads. The CG model aims at reproducing specific features of the high-resolution model, such as structure or thermodynamics [7–9]. The reduced representation eliminates some molecular friction, smoothens the energy landscape, and thereby effectively accelerates sampling of the conformational space.

While mapping from fine to coarse is straightforward, going the reverse way is no trivial task. Backmapping means reintroducing lost degrees of freedom: from CG beads to atoms. The reduced CG resolution implies that one CG configuration will correspond to an *ensemble* of atomistic microstates. Ideally, the CG model should perfectly reproduce the Boltzmann distribution of the atomistic system along the CG degrees of freedom—the many-body potential of mean force. As such, backmapping aims at generating an atomistic structure drawn from the probability distribution of atomistic microstates, given the CG configuration.



The general strategy of existing backmapping schemes is to insert an initial set of atomistic coordinates into the CG structure [10]. A generic approach is to randomly place the atoms close to their corresponding CG bead center [11, 12]. To relax the initial atomistic structure energy minimization is required and a subsequent MD simulation has to be performed to equilibrate the system to obtain the correct Boltzmann distribution. The computational cost for the subsequent minimization and equilibration procedures can become significant for high-dimensional systems. This is also true for backmapping large numbers of CG configurations [13]. Furthermore, generating the initial atomistic structure often requires human intuition to avoid trapping in local minima. For example, the protocol of Wassenaar *et al* needs to introduce geometric modifiers to correctly reproduce the distribution of torsion angles in phospholipids [12].

Another, more specific, approach is to superimpose presampled fragments from a correctly sampled distribution of all-atom structures onto the CG configuration [1, 14–16]. In this case the effort of subsequent energy minimization and MD simulation can be reduced dramatically, but are still required to reproduce the desired Boltzmann distribution.

In this work we introduce DeepBackmap, a backmapping scheme based on deep convolutional neural networks (deep CNNs). Our method can be used to predict equilibrated atomistic structures directly from the CG configuration without running any energy minimization or MD simulation. Our approach works well even for a limited amount of training data making it advantageous to study systems that are expensive to sample. This is achieved using generative adversarial networks (GANs) [17–19], a particular type of generative model based on deep networks: During training, an auxiliary critic learns a distance metric between generated and training data. While the critic is trained to maximize the distance, the objective of the generator is to minimize it.

To extend GANs to a conditional model, an auxiliary input can be introduced to both, the generator and the critic, which is taken to be the conditional variable [20, 21]. Here we use conditional GANs to learn a coarse-to-fine mapping that re-introduces degrees of freedom with the correct conditional statistical weight. To this end, we use the CG structure as an auxiliary input.

While generating low-energy geometries for molecular compounds remains a challenge that is still tackled largely by MD simulations, machine learning (ML) approaches receive growing attention. Recent approaches using ML include autoregressive models [22, 23], invertible neural network [24], Euclidean distance matrices [25], graph neural networks [26], and variational auto-encoders (VAEs) [27]. Intuitively, VAEs are in particular interesting for the reverse-mapping task, since they can be used to not only learn the CG variables but also the decoding back to atomistic detail, as it was shown by Wang and Gómez-Bombarelli [27]. Here instead, we focus on the last step for a pre-existing CG model. Backmapping involves the reintroduction of degrees of freedom, and thereby a single CG configuration will map to many atomistic microstates. Therefore the model needs to learn to generate this (Boltzmann-distributed) collection of microstates, using noise and complex statistical dependencies that are conditional on the CG configuration. Prior work has shown the benefits of adversarial training when aiming at high-quality generative tasks [28].

Our study uses a convolutional GAN, which has shown the ability to model highly complex and detailed probability distributions (statistical dependency structures) in computer vision applications [29]. The seemingly unintuitive training protocol of GANs circumvents the hard problem of fitting the posterior distribution to training data. Instead of explicitly learning the distribution, they only tune a sampler (the generator) to produce samples indistinguishable from the training distribution (for the critic). For high-dimensional data sets, such as the joint distribution of many atoms in molecules, previous methods become either intractable or lose resolution, dependencies, or both.

The convolutional network architecture requires a regular discretization of 3D space, prohibiting scaling to larger spatial structures. We therefore combine the convolutional generator with an autoregressive approach that, in an outer loop, reconstructs the fine-grained structure incrementally, atom by atom. In each step, we use only local information, making the method scalable to arbitrary system sizes and applicable to condensed phase systems.

We test our approach on a dense polymeric system: syndiotactic polystyrene (sPS). sPS not only shows complex structural features in the amorphous melt, it can also crystallize [30–32]. An illustration of the CG and the atomistic representation of the molecule can be found in figure 1.

A thorough investigation of soft-matter systems often requires the consideration of different state points. The temperature transferability of the model is thereby an important aspect. Similar to the work of Brasiello *et al* [16], where a database of all-atom structures sampled in a melt is used to reverse-map CG structures at lower temperatures, we train our model solely on data obtained in a high-temperature melt and probe the model's transferability to lower temperatures. Remarkably, the model still performs well through the phase transition into the crystalline phase. This indicates that the microscopic degrees of freedom learned by the model have weak temperature dependence: local information is relatively insensitive to the state point, while the larger-scale features are captured by the CG structure.

2. ML model

In the following, we discuss our approach. We start with a description of the molecular simulation scenario our method handles, and then discuss in detail how the deep backmapping algorithm works.

2.1. Setup

We define notation for the CG and atomistic resolutions, as well as the backmapping procedure:

CG resolution Let $\{\mathbf{A}_I = (\mathbf{R}_I, C_I) | I = 1, \dots, N\}$ denote the set of N coarse grained beads. Each bead has position $\mathbf{R}_I \in \mathbb{R}^3$ and bead type C_I .

Atomistic resolution Let $\{\mathbf{a}_i = (\mathbf{r}_i, c_i) | i = 1, \dots, n\}$ denote the set of n atoms, with position $\mathbf{r}_i \in \mathbb{R}^3$ and atom type c_i . We denote $\varphi_I \subset \{\mathbf{a}_i | i = 1, \dots, n\}$ as the set of atoms contained in the CG bead \mathbf{A}_I .

Backmapping Backmapping requires us to generate a set of n atom positions $\mathbf{r}_1, \dots, \mathbf{r}_n$ conditional on the CG structure, given by the N beads $\mathbf{A}_1, \dots, \mathbf{A}_N$, as well as the atom types c_1, \dots, c_n . We express this problem as a conditional probability $p(\mathbf{r}_1, \dots, \mathbf{r}_n | c_1, \dots, c_n, \mathbf{A}_1, \dots, \mathbf{A}_N)$.

We now propose a ML technique that takes examples of corresponding coarse- and fine-grained examples as input and from this *training data* learns the conditional distribution p . Specifically, we do not learn p directly, which is well-known to be a hard problem for high-dimensional phase spaces [17], but rather infer a sampler that can generate further samples from p , see figure 2.

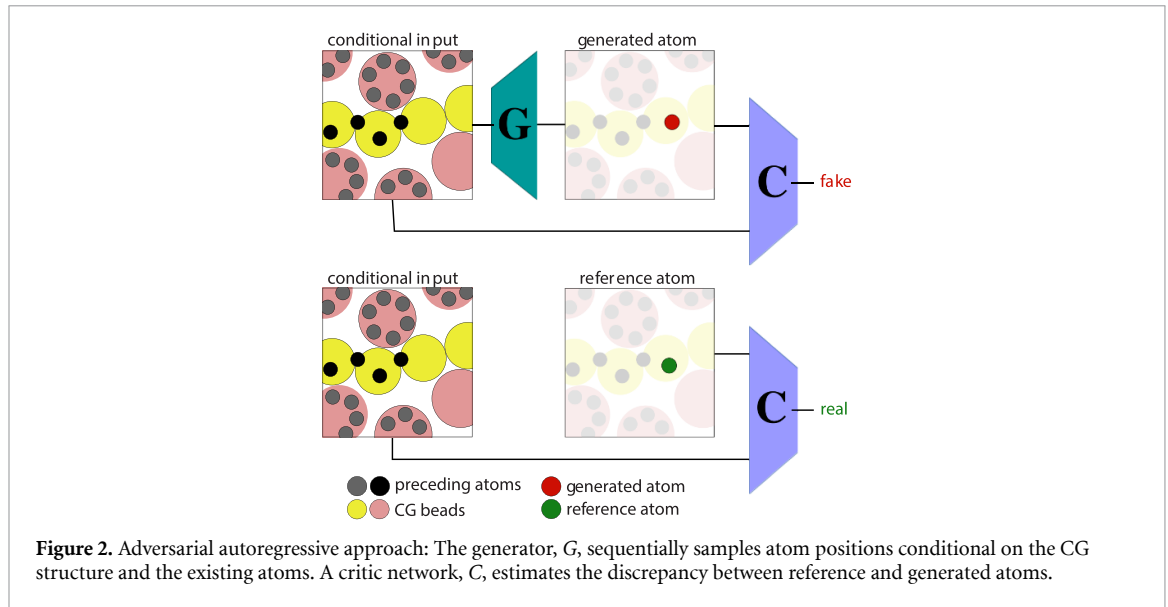
2.2. Outer loop: autoregressive model

Rather than learning to sample from $p(\mathbf{r}_1, \dots, \mathbf{r}_n | c_1, \dots, c_n, \mathbf{A}_1, \dots, \mathbf{A}_N)$ directly, we propose to factorize p in terms of atomic contributions, where the generation of one specific atom becomes conditional on both CG beads as well as all the atoms previously reconstructed [23]. Based on this factorization we can train a generative network, G , to generate and refine the atom positions sequentially.

The backmapping scheme hereby consists of two steps: (i) An initial structure is generated using the factorization

$$p(\mathbf{r}_1, \dots, \mathbf{r}_n | c_1, \dots, c_n, \mathbf{A}_1, \dots, \mathbf{A}_N) = \prod_{i=1}^n p(\mathbf{r}_{S(i)} | \mathbf{r}_{S(1)}, \dots, \mathbf{r}_{S(i-1)}, c_{S(1)}, \dots, c_{S(i)}, \mathbf{A}_1, \dots, \mathbf{A}_N), \quad (1)$$

where S sorts the atoms in the order of reconstruction and $\{\mathbf{r}_{S(1)}, \dots, \mathbf{r}_{S(i-1)}\}$ correspond to atoms that have been already reconstructed. The dependence on earlier predictions of G makes our approach *autoregressive*. This procedure would be exact in a Markovian regime where each atom interacts directly only with its predecessor and successor (so-called 'chain structures' [33]). Unfortunately the complexity of condensed-phase liquids calls for more feedback to avoid steric clashes; (ii) Intuitively, we cannot optimally



place an atom without its whole environment present. This issue is compounded for ring-like structures, like the phenyl group in polystyrene. To this end we perform a variant of Gibbs sampling, which iteratively resamples along the sequence S several times [34]. Each further iteration still updates one atom at a time, but uses the knowledge of *all* other atoms. Experiments confirmed that such Gibbs sampling leads to a good approximation of p , even with a small number of iterations and fixing the atom ordering.

2.3. Representation

Iterative sampling algorithms, such as the Gibbs sampler, have high computational cost. We hereby optimize our approach by means of a robust learning algorithm that can capture complex dependencies in the local environment directly.

The problem of learning complex, high-dimensional and high-order dependencies in generative models has received considerable attention in computer vision. The most successful technique for this task are generative deep CNNs [35] trained by adversarial training [17, 29]. There is also growing evidence that deep networks are also effective in capturing the statistics of physical systems [23, 24].

In order to leverage deep CNNs for our task, an explicit spatial discretization of the ambient space, similar to pixels in an image, is required. The standard technique is to use a voxel-based representation [36]. To this end, we represent atoms and CG beads with a smooth density, $\gamma(\mathbf{x})$ and $\Gamma(\mathbf{x})$, respectively.

The particle densities are modeled using Gaussian distributions, such that for atom i we define

$$\gamma_i(\mathbf{x}) = \exp\left(-\frac{(\mathbf{x} - \mathbf{r}_i)^2}{2\sigma^2}\right), \quad (2)$$

where \mathbf{x} is the spatial location in Cartesian coordinates, expressed on a discretized grid due to the voxel representation. The density is centered around particle position \mathbf{r}_i with Gaussian width σ , treated as a hyperparameter. CG beads are similarly represented.

2.3.1. Locality

The high costs of large regular 3D grids are the reason for employing deep CNNs only locally and using the previously described outer loop to build-up larger structures incrementally using autoregressive sampling. To make the model scalable to large system sizes, we assume locality by limiting the information about the environment to a cutoff r_{cut} .

We encode the local environment of an atom i or CG bead I by means of the density of particles placed around it, denoted $\xi_{i,I}$ and Ξ_I , respectively. We sum over all atoms or beads within a cubic environment of size $2r_{\text{cut}}$. We shift all atom and bead positions around the CG bead of interest, I . Further, we rotate the local environment to a local axis system. This improves generalization from limited training examples by removing three translational and two of the rotational degrees of freedom, i.e. the ML algorithm does not need to learn the corresponding coordinate invariance from (additional) examples.

Specifically, we align the bond between consecutive CG beads $I - 1$ and I to the local z axis using a rotation matrix M_I to construct the local environment of atom i

$$\xi_{i,I}(\mathbf{x}) = \sum_{j=0}^{i-1} \gamma_{S(j)}(M_I(\mathbf{x} - \mathbf{R}_I)), \quad (3)$$

which extends over the region $-r_{\text{cut}} < x_\alpha < r_{\text{cut}}$ and α runs over the three Cartesian coordinates. Similarly the CG environment is constructed as

$$\Xi_I(\mathbf{x}) = \sum_{J=0}^N \Gamma_J(M_I(\mathbf{x} - \mathbf{R}_I)). \quad (4)$$

In this work we set $r_{\text{cut}} = 6 \text{ \AA}$, such that several CG beads are included in the local environment (see supplementary figure S2 (<https://stacks.iop.org/MLST/1/045014/mmedia>)). Importantly, $\xi_{i,I}$ and Ξ_I are discretized on a regular grid.

2.3.2. Feature embedding

A convolutional neural network (CNN) takes an image (typically 2D or 3D) as input where every pixel or voxel is vector-valued. For example, an RGB image consists of three *feature channels*: One channel for every primary color. Here, we store a number of feature channels in each voxel that represent the presence of other atoms or beads of a certain kind. In the most basic version, we could use a single feature channel to encode all other atoms. However, this would make it impossible to distinguish their type and might also lead to clutter. The opposite extreme would be to assign a separate feature channel to each atom. The downside here is not only increased memory costs but, more importantly, the loss of permutation invariance of the atoms.

As shown in figure 3(a), we create separate feature channels for each atom type. Atom types are distinguished not only by element but additionally by chemical similarity, i.e. atoms of a given type can be treated as identical in the MD simulation. Specifically, we classify similarity following the force field for sPS by Mueller-Plathe [37]. For atoms of the same type, we further add channels to distinguish the functional form of interaction to the current atom of interest. Interaction types distinguish between bond, bending angle, torsion, and Lennard-Jones. Similarly, we use separate channels to encode the different CG bead types.

Formally, let $f \in \{1, 2, \dots, N_F\}$ denote the index of the N_F different feature channels. We define the activation function, $h_f(S(j))$, to denote association with a channel f

$$h_f(S(j)) = \begin{cases} 1, & \text{if atom } S(j) \text{ has feature } f \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

and $H_f(J)$ to similarly encode the bead type. We then build a density map for each channel for both atomic environments

$$\xi_{i,I}(\mathbf{x}, f) = \sum_{j=0}^{i-1} \gamma_{S(j)}(M_I(\mathbf{x} - \mathbf{R}_I)) h_f(S(j)), \quad (6)$$

and CG environments

$$\Xi_I(\mathbf{x}, f) = \sum_{J=0}^N \Gamma_J(M_I(\mathbf{x} - \mathbf{R}_I)) H_f(J). \quad (7)$$

2.4. Generative model

Training a generative model is challenging as it requires to measure and optimize closeness of the target distribution and the generated distribution of the model. A direct maximum likelihood training, where the model's parameters are tuned such that the likelihood of observing the data given the model is optimized, is infeasible in high dimensions because the normalization factor—the partition function—cannot be computed efficiently.

Approaches to circumvent these limitations include approximate techniques like variational autoencoders, where a stochastic lower bound of the log-likelihood is optimized. Another solution are likelihood-free methods, such as adversarial training [17], that operate indirectly, by building a sampler and comparing its output to actual data with a second, 'adversarial' network. In recent literature, this approach appears to yield the strongest results, in particular on high-dimensional and hard to model image spaces

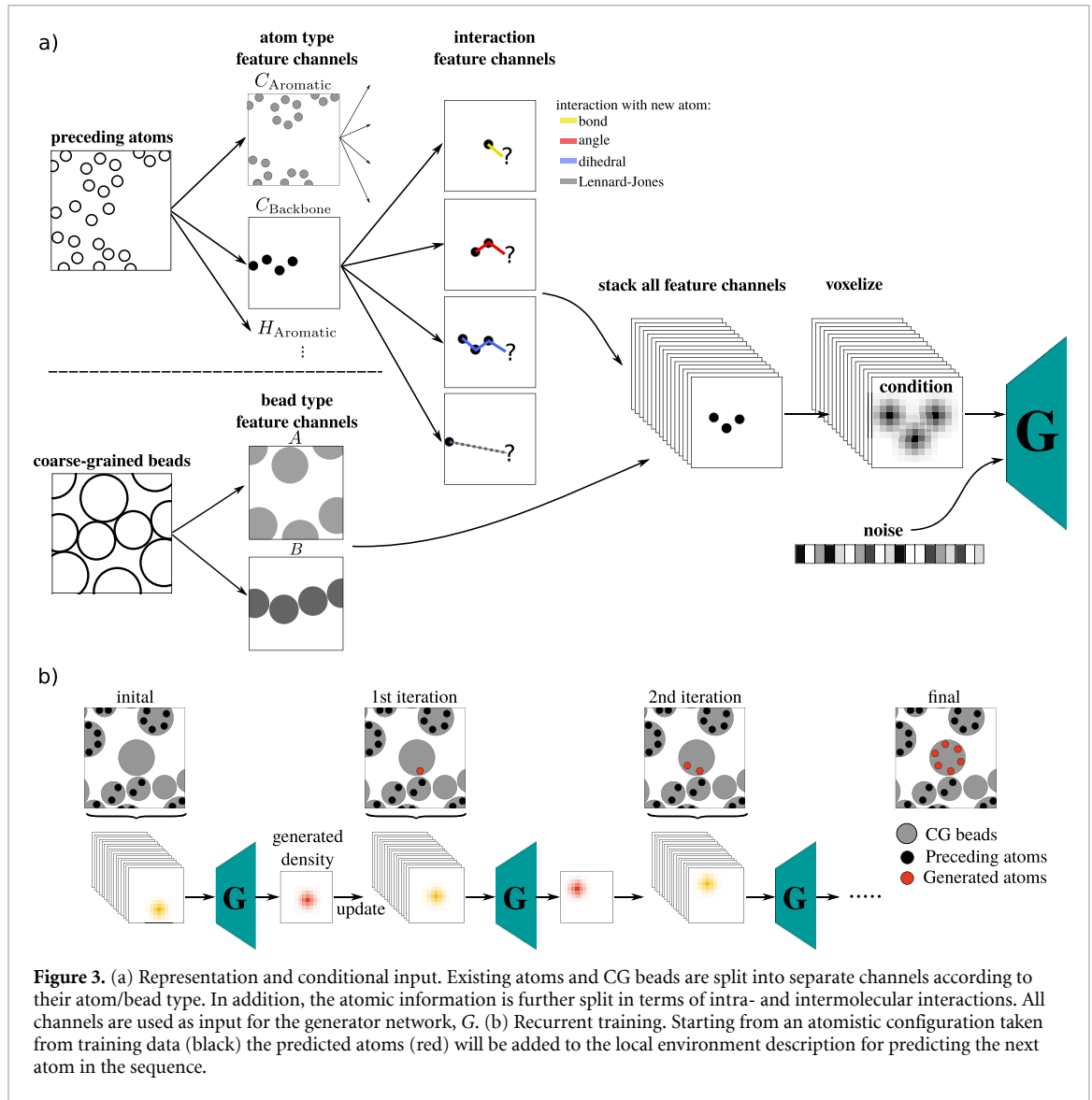


Figure 3. (a) Representation and conditional input. Existing atoms and CG beads are split into separate channels according to their atom/bead type. In addition, the atomic information is further split in terms of intra- and intermolecular interactions. All channels are used as input for the generator network, G. (b) Recurrent training. Starting from an atomistic configuration taken from training data (black) the predicted atoms (red) will be added to the local environment description for predicting the next atom in the sequence.

[29]. The next best option are auto-regressive models, which tackle the complexity issue by learning single decisions at a time [22]. We use this approach in the outer loop but employ the more expressive GAN for modeling the local placement of atoms.

Formally, to perform adversarial training, a second network is introduced, called critic C , to distinguish between training samples and samples from the generative model G . The generator competes with the critic C and is trained to generate samples that C can not distinguish anymore from training samples. In the conditional adversarial framework [20, 21] both networks G and C are provided with auxiliary information like a class label to generate samples related to this information. In this study, we use a conditional generative adversarial network to generate new atom positions from a random noise vector $z \sim N(0, 1)$ and the conditional input $u_i := \{\xi_{i,I}, \Xi_I, c_i\}$ consisting of the local environment representation $\xi_{i,I}$ and Ξ_I , as well as the current atom type c_i . In a first step, the generator G predicts a smooth-density representation $\hat{\gamma}_i := G(z, u_i)$.

2.4.1. From densities to coordinates

While the smooth-density representation $\hat{\gamma}_i$ is adequate for a CNN, we ultimately wish to collapse these back to point coordinates. We simply compute a weighted average, discretized over the voxel grid

$$\hat{r}_i = \int d\mathbf{x} \hat{\gamma}_i(\mathbf{x}) \approx \sum_m \sum_k \sum_l x_{mkl} \hat{\gamma}_i(x_{mkl}). \quad (8)$$

This step is performed for each generated density separately, one atom at a time. We note that this density-collapse step is differentiable and can thus be easily incorporated in a loss function.

2.4.2. Training

Training of a GAN model is split in two networks: the adversarial critic and the generative network. The following describes the two loss functions.

We train a critic network C to distinguish between reference densities γ_i related to the conditional input $u_i = \{\xi_{i,I}, \Xi_i, c_i\}$ and generated densities $\hat{\gamma}_i = G(u_i, z)$. The critic aims at both (i) distinguishing reference from generated samples and (ii) ensuring smoothness of the classification with respect to the generator's parameters. Both criteria can be fulfilled using a variant of adversarial models where the critic C is used to approximate the Wasserstein distance [18].

The loss function is constructed using the Kantorovich-Rubinstein duality, which requires C to be constrained to the set of 1-Lipschitz functions. A differentiable function is 1-Lipschitz if and only if it has gradients everywhere with norm at most one. A soft version of this constraint is enforced with a penalty on the gradient norm [19]

$$\mathcal{L}_C = \mathbb{E}_i \left[C(u_i, \gamma_i) - C(u_i, G(u_i, z)) + \lambda_{\text{gp}} (\|\nabla_{\tilde{u}_i, \tilde{\gamma}_i} C(\tilde{u}_i, \tilde{\gamma}_i)\|_2 - 1)^2 \right], \quad (9)$$

where $(u_i, \tilde{\gamma}_i)$ is interpolated linearly between pairs of points (u_i, γ_i) and $(u_i, G(u_i, z))$. The prefactor λ_{gp} scales the weight of the gradient penalty.

For the generator we combine two aspects to help generate faithful structures: (i) the critic that compares reference and generated samples, $C(u_i, G(u_i, z))$, and (ii) a physical prior, Φ . Φ aims at accelerating convergence by helping the generator refine its output. It combines both force-field-based energy contributions, E_{FF} , and a geometric center-of-mass distance contribution, d_{COM} . The prior depends on the set of atoms corresponding to a CG bead, φ_I for reference atoms and $\hat{\varphi}_I$ for generated atoms, as well as reference atoms N_I in the local neighborhood of different beads:

$$\Phi(\varphi_I, \hat{\varphi}_I, N_I) = E_{\text{FF}}(\varphi_I, \hat{\varphi}_I, N_I) + d_{\text{COM}}(\varphi_I, \hat{\varphi}_I). \quad (10)$$

The force-field-based term penalizes discrepancies between samples with respect to specific intra- and intermolecular interactions within all neighborhoods N_I .

$$E_{\text{FF}}(\varphi_I, \hat{\varphi}_I, N_I) = \sum_t |\varepsilon_t(\varphi_I, N_I) - \varepsilon_t(\hat{\varphi}_I, N_I)|, \quad (11)$$

where t runs over the interaction types: intramolecular bond, angle, and dihedral, and non-bonded Lennard-Jones. The set of interactions follow the reference atomistic force field. In the following, let $\theta_I = \{i | a_i \in \varphi_I\}$ be the set of atom indices for atoms contained in φ_I . The second term in the physical prior penalizes discrepancies in the center-of-mass geometry between samples

$$d_{\text{COM}}(\varphi_I, \hat{\varphi}_I) = |g(\varphi_I) - g(\hat{\varphi}_I)|, \quad (12)$$

where g refers to the center of mass

$$g(\varphi_I) = \frac{\sum_{i \in \theta_I} m_i \mathbf{r}_i}{\sum_{i \in \theta_I} m_i}, \quad (13)$$

with m_i being the mass of atom a_i .

Overall this leads to the following loss function for the generator

$$\mathcal{L}_G = \mathbb{E}_I \left[\mathbb{E}_{i \in \theta_I} [C(u_i, G(u_i, z))] + \lambda_{\Phi} [\Phi(\varphi_I, \hat{\varphi}_I, N_I)] \right] \quad (14)$$

where the prefactor λ_{Φ} scales the weight of the physical prior.

The two loss functions, \mathcal{L}_C and \mathcal{L}_G are trained iteratively and alternately until the process reaches equilibrium. Monitoring the training process of a GAN is difficult, since it is not possible to measure the accuracy of the model. As an alternative we turn to the evolution of the adversarial loss for the generator/critic, which is known to correlate with the sample quality and therefore is a useful measure for the training progress [19]. Furthermore, we keep track of the force-field-based loss E_{FF} and the geometric term d_{COM} . The corresponding plots can be found in figure S7.

2.4.3. Implementation details

We choose a 3D CNN architecture with residual connections for G and C [38]. See figure S6 for a detailed network description. The evolution of the different loss terms can be found in figure S7.

The model is trained for 38 660 steps (≈ 25 epochs) in total using a batch size of 36. For stability reasons, we start training with $\lambda_{\Phi} = 0$ and increase it smoothly to $\lambda_{\Phi} = 0.01$ from step 6000 to 10 000. Training is performed using the Adam optimizer with learning rates $5 \cdot 10^{-5}$ for the generator and 10^{-4} for the critic. The prefactor scaling the weight of the gradient penalty term is set to $\lambda_{gp} = 0.1$. To obtain reliable gradients for the generator, the critic should be trained until optimality. Therefore the critic C is trained five times in each iteration while the generator G is trained just once.

We train the model recurrently on atom sequences containing either all heavy (carbon) or light (hydrogen) atoms corresponding to a single CG bead. During training, the initial atomistic environment representation $\xi_{i,I}$ for each sequence is generated from training data and contains the atoms present (according to the order S) in the local neighborhood N_I of bead I . After each step, the generated atom density is added to the local environment representation for the next atom in the sequence, as illustrated in figure 3(b), until all atoms of the sequence are generated.

In the Gibbs-sampling step, information of all preceding and subsequent atoms is used to refine the positions of light atoms. On the other hand for heavy atoms we remove hydrogens from the current and adjacent beads such that misplaced hydrogens will not hinder G to find suitable positions for the heavy atoms.

Note that our architecture is not fully rotationally equivariant. While promising work has been done to design rotationally equivariant networks [39, 40], extending these approaches to generative tasks is not straightforward. Here instead, we reduce the rotational degrees of freedom by aligning the region considered by the generator according to the position of the central bead and the difference vector to the previous bead. This leaves one rotational degree of freedom around the director axis, for which we augment the training set by means of rotations. During prediction we feed different orientations about said axis and choose the structure with the lowest energy from the generated ensemble.

3. Computational methods

3.1. Reference data

The atomistic data in this study was reported in Liu *et al* [32]; the underlying force field is based on the work of Mueller-Plathe [37]. Replica Exchange MD simulation, a temperature-based enhanced sampling technique, was used to sample the system. All simulations were performed using the MD package GROMACS 4.6 [41]. MD simulations are performed in the NPT ensemble using the velocity rescaling thermostat and the Parrinello-Rahman barostat. An integration timestep of 1 fs is used. For additional details regarding the simulations the reader is referred to the work of Liu *et al* [32].

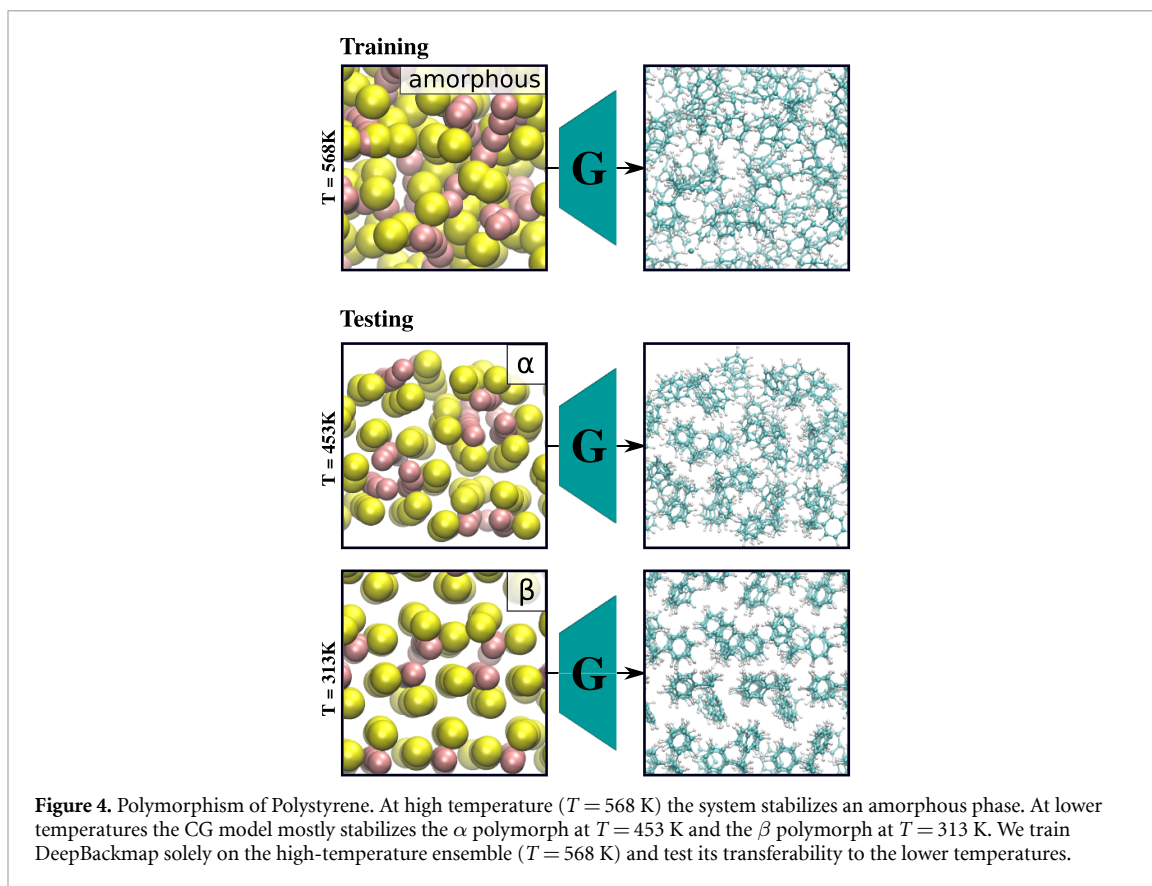
Our training/test data consists of pairs of corresponding fine- and CG snapshots. To this end, we start from the atomistic frame and apply a fine-to-coarse mapping to obtain the CG structures. We use uncorrelated snapshots from three different trajectories simulated at $T = 568$ K, 453 K, and 313 K. The system includes 36 polystyrene chains and each chain consists of 10 monomers.

The fine-to-coarse mapping is based on the CG model developed by Fritz *et al* [31]. It represents the CG molecule as a linear chain, where each monomer is mapped onto two CG beads of different types, denoted A for the chain backbone and B for the phenyl ring (see figure 1). Bonds are created between the backbone beads A-A and between backbone and phenyl ring beads A-B. The coarse grained model, parameterized in the melt, is transferable to the crystalline phase and stabilizes the experimentally observed α and β polymorphs [32].

3.2. Baseline model

We compare our results with a generic backmapping scheme developed by Wassenaar *et al* [12]. This method places each particle on the weighted average position of the coarse grained beads it belongs to and optionally adds a random displacement. The protocol continues with corrections to the structure using geometric modifiers, setting the alignment of the next particle as cis, trans, out, or chiral with respect to the others. Note that those modifiers need first be manually defined by the user.

The corrected structure is then relaxed by a force-field based energy minimization. The first cycle of energy minimization consists of 200 steps and is performed with non-bonded interactions turned off. The second cycle of energy minimization consists of 5000 steps with all interactions turned on. Clearly the energy minimized structures will not capture the right Boltzmann distribution and therefore the protocol of Wassenaar continues with several cycles of position restrained MD simulations. Since we aim for a backmapping scheme that performs well without running molecular dynamics simulation, we stop the



protocol after the energy minimization and compare the methods without running any further MD simulations.

4. Results

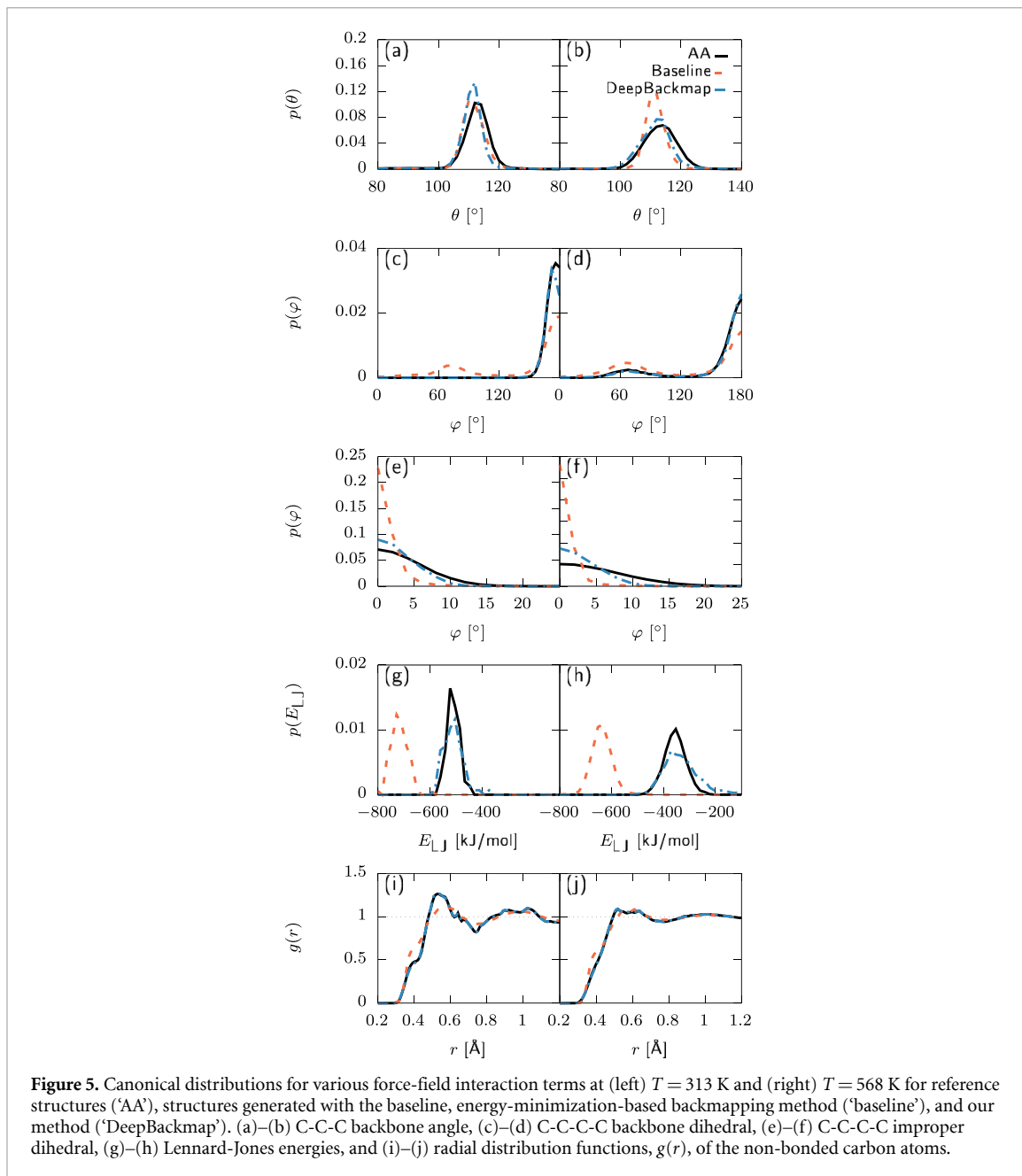
We apply DeepBackmap to a challenging condensed-phase molecular system: syndiotactic polystyrene. Despite its simple chemical structure, polystyrene displays a rich conformational space. Its syndiotactic form can crystallize, and exhibits complex polymorphic behavior. Upon thermal annealing, a polystyrene melt undergoes a phase transition from amorphous to a crystalline phase at $T \approx 450$ K. The CG model was shown to stabilize the two main crystal polymorphs α and β (see figure 4) [32].

We probe the model's ability to transfer across temperatures. To this end, we train DeepBackmap solely on high-temperature, amorphous configurations, but validate it at several temperatures (figure 4). The training set consisted of only 12 snapshots simulated at $T = 568$ K. The model was then applied to MD configurations at $T = 568$ K, 453 K, and 313 K, each containing 78 samples that were not used during training. For brevity we only report results about the highest and lowest temperature. We evaluate the performance of the model regarding its ability to reproduce structural and energetic features of the reference atomistic configurations, as well as a comparison with the baseline method.

4.1. Local structural and energetic features

Figure 5 shows distribution functions for structural and energetic properties. We first analyze the hold-out validation data at $T = 568$ K (right column), the temperature at which DeepBackmap was trained on. Our method generates configurations that are remarkably close to the reference Boltzmann distribution ('AA'), especially when considering the current state of the art. The distributions of intramolecular carbon backbone angle and dihedral show very good agreement (figures 5(a)–(d)). On the other hand, the baseline method displays too narrow distributions and spurious peaks. While the distribution for the carbon improper dihedral of the aromatic structure is slightly too narrow, we emphasize the small range of angles (figures 5(e)–(f)), due to the imposed planarity of the ring. The baseline method significantly suppresses fluctuations around the planar structure.

The Lennard-Jones energies shown in figures 5 (g)–(h) obtained for each chain separately also match remarkably well with the reference distribution—this aspect is of tremendous importance to generate well-equilibrated structures in a condensed environment. We do observe slightly large high-energy tails,



often due to an accumulation of errors of misplaced atoms impacting the subsequent placement of neighbors in our autoregressive approach. On the other hand, the baseline model systematically and drastically over-stabilizes the system. This stems from the energy minimization itself, which does not account for the diversity of microstates at a specific canonical state point. Practically this is commonly resolved by applying a thermostat to recover the right state point. Here, because we aim at a comparison between backmapping schemes *prior* to any MD, we omit this last step.

4.2. Transferability to low temperatures

While we fix the original training of DeepBackmap to the high-temperature ensemble, we hereby test it at low temperature ($T = 313$ K), *without* reparametrization. Beyond a mere shift in temperature, the system undergoes a phase transition, going from an amorphous phase to a crystalline state with different polymorphs. The distributions in figures 5(a)–(g) (left column) show remarkably accuracy: DeepBackmap retains its performance displayed for the training temperature. Upon cooling the distributions do show a number of significant changes: narrower distribution in the angle, vanishing of the side peak in the backbone dihedral, and large shift of the Lennard-Jones energies.

The transferability of DeepBackmap is highlighted when compared to the baseline model, which retains much of its features found at high temperature. This is especially apparent for the side peak of the backbone dihedral.

4.3. MD simulation

Backmapped structures are often used as starting points for further MD simulations. All backmapping schemes rely on some amount of energy minimization—some need very little thanks to a good starting structure, but others require lengthy preparations including temperature ramp-up phase and thermostat/barostat equilibration. Here we use the latter case as a baseline test to compare to our methodology.

Figure SI.7 displays the evolution of the potential energy during MD simulations *without* heat-up at $T = 313$ K starting from structures generated with the different methods. Initial velocities are generated according to a Maxwell distribution. The evolution of the potential energy of structures generated with DeepBackmap follows closely the evolution of reference AA structures. The potential energy reaches a steady value after 100 ps. On the other hand, energy minimized structures from the baseline method settle at significantly higher energies indicating badly initialized structures that get trapped into local minima with high energy barriers.

Last, we compare the computational cost of the two methods. Running the full baseline method—including a 1 ns MD run with thermostat to recover the right canonical state point—on an Intel(R) Core(TM) i7-7700 CPU 3.60 GHz takes roughly 1300 s for each configuration. The wall-clock time using our method on an Intel(R) Xeon(R) Gold 6138 CPU 2.00 GHz together with a Tesla Volta V100 GPU is approximately 2000 s. The time spent on the GPU is roughly 3/4 of the total wall-clock time. The two methods are therefore roughly on par. We further emphasize that our current implementation is only a proof of concept and has not been optimized for efficiency yet, both regarding the data processing and network architecture.

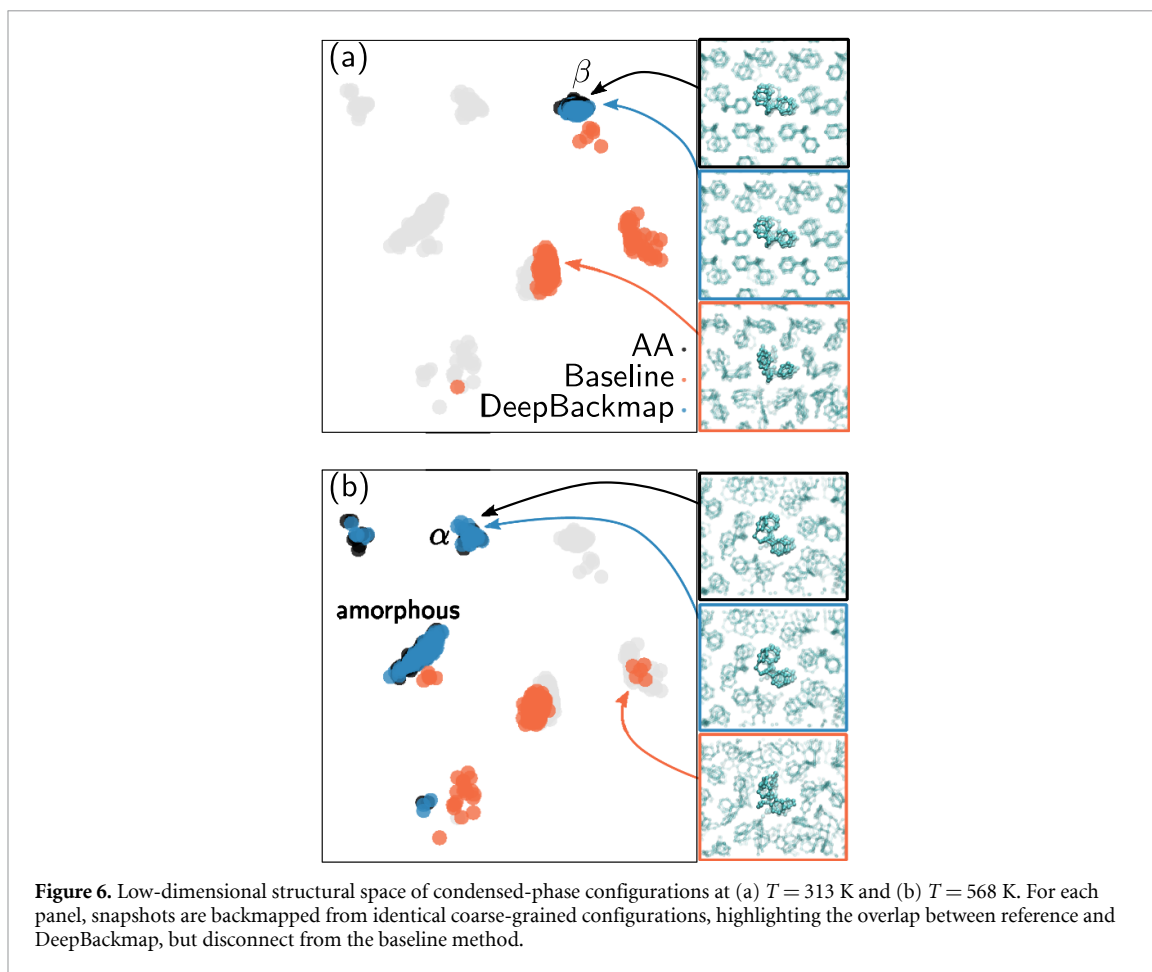
4.4. Large-scale structural features

To further evaluate the large-scale structural features, we turn to the pair correlation function, $g(r)$. Figures 5(i)–(j) focuses here on non-bonded carbon pairs. We can see an excellent agreement between the reference AA $g(r)$ and the DeepBackmap results for *both* temperatures. This clearly indicates that the local packing of the polystyrene chains is well reproduced, even for different state points that were not used during training. As expected the baseline method does not reproduce the pair correlation satisfyingly, especially fails in the crystalline phase.

Beyond the pair statistics, we wish to probe the accuracy of the reconstruction at higher order. We build a two-dimensional map representing proximity relationships between condensed-phase structures. We focus on the local environment around each backbone carbon atom that directly links to a side chain (i.e. every other backbone carbon). The pairwise distance between two such environments is encoded using a similarity kernel based on SOAP representations [42]. Hydrogens are ignored from the representation. To compare two structures we compute a covariance matrix containing all the pairwise distances between atomic environments, followed by a regularized entropy match kernel [43]. We further apply Sketchmap to obtain a reduced-dimensional projection of the conformational space [44, 45]. Figure 6 displays a number of clusters that correspond to different environments. The low-temperature reference data shows a single cluster (figure 6(a) in black), corresponding to the β phase, while the high-temperature reference shows more diversity (i.e. α , amorphous phase, and others). DeepBackmap overlaps significantly with the reference points at both temperatures, highlighting the high structural fidelity. This is not the case for the energy-minimized structures of the baseline model, as they cover different areas of the low-dimensional map. The baseline also fails to reproduce the correct number of clusters: at both temperatures the baseline model displays three to four clusters, highlighting a lack of temperature sensitivity.

5. Conclusions

In this study we propose a new backmapping scheme based on deep neural networks. The model inserts atomistic details based on large-scale structures from a CG snapshot. To this end we use a conditional generative network where the CG information is used as an auxiliary input. We train our model, DeepBackmap, combining an adversarial loss function with a physical prior. The method is scalable to arbitrary system sizes since only local information is used. Our method is able to generate well-equilibrated high-resolution structures of condensed-phase systems. Critically, and unlike current methods, our approach does not need MD (MD) simulations to yield the correct Boltzmann distribution.



We applied our methodology to a complex condensed-phase system made of syndiotactic-polystyrene chains. The model displays remarkable transferability properties: while trained solely on high-temperature melt configurations, DeepBackmap performs well at significantly low temperatures, where the system is in a crystalline state. This indicates that the local correlations learned by the model are transferable across different state points, aided by the physics we incorporated into the GAN.

We rationalize these remarkable features in terms of scale separation: the large-scale features are encoded in the CG configurations, while the model only need to generate equilibrated *local* correlations. Local features are less affected by temperature, since the underlying covalent interactions operate primarily on an energy scale significantly larger than $k_B T$. As such the backmapping operates on two different sources of information: (i) the conditional CG configurations and (ii) the learned local correlations. Most of the temperature dependence is carried by the former, such that DeepBackmap can accurately produce an accurate Boltzmann distribution across a phase transition from training at a single (high) temperature. On the other hand, it is not clear that training only at low temperature—especially below a phase transition—should lead to satisfying transferability at high temperatures, given the larger, broader conformational space spanned.

Beyond the evident advantages of generating equilibrated molecular structures, our approach offers the perspective of a tighter integration of multiscale models: The information of the CG is efficiently recycled into the higher resolution. Avoiding unnecessary equilibrations upon upscaling will help connect models at different scales—an important task at the dawn of the exascale computing era.

Acknowledgments

The authors thank Kiran H Kanekal, Yasemin Bozkurt Varolgüneş, and Arghya Dutta for critical reading of the manuscript. We are grateful to Chan Liu for providing coarse-grained and atomistic simulations of syndiotactic polystyrene. This work was supported in part by the TRR 146 Collaborative Research Center of the Deutsch Forschungsgemeinschaft as well as the Max Planck Graduate Center. TB acknowledges financial support by the Emmy Noether program of the Deutsche Forschungsgemeinschaft (DFG). Part of this

research was performed while MS and TB were visiting the Institute for Pure and Applied Mathematics (IPAM), which is supported by the National Science Foundation (Grant No. DMS-1440415).

ORCID iD

Tristan Bereau  <https://orcid.org/0000-0001-9945-1271>

References

- [1] Peter C and Kremer K 2009 Multiscale simulation of soft matter systems—from the atomistic to the coarse-grained level and back *Soft Matter* **5** 4357–66
- [2] Bereau T, Andrienko D and Kremer K 2016 Research Update: Computational materials discovery in soft matter *APL Mater.* **4** 053101
- [3] Mitsutake A, Mori Y and Okamoto Y 2013 Enhanced sampling algorithms *Biomolecular Simulations* (Berlin: Springer) pp 153–95
- [4] Shaw D E et al 2009 Millisecond-scale molecular dynamics simulations on anton *Proc. of the Conf. on High Performance Computing Networking, Storage and Analysis* pp 1–11
- [5] Kremer K and Müller-Plathe F 2002 Multiscale simulation in polymer science *Mol. Simul.* **28** 729–50
- [6] Horstemeyer M F 2009 Multiscale modeling: a review *Practical Aspects of Computational Chemistry* (Berlin: Springer) pp 87–135
- [7] Voth G A 2008 *Coarse-Graining of Condensed Phase and Biomolecular Systems* (Boca Raton, FL: CRC press)
- [8] Noid W G 2013 Perspective: Coarse-grained models for biomolecular systems *J. Chem. Phys.* **139** 090901
- [9] Brini E, Algaer E A, Ganguly P, Li C, Rodríguez-Ropero F and van der Vegt N F 2013 Systematic coarse-graining methods for soft matter simulations—a review *Soft Matter* **9** 2108–19
- [10] Tschöp W, Kremer K, Hahn O, Batoulis J and Bürger T 1998 Simulation of polymer melts. ii. from coarse-grained models back to atomistic description *Acta Polym.* **49** 75–9
- [11] Rzepiela A J, Schäfer L V, Goga N, Risselada H J, De Vries A H and Marrink S J 2010 Reconstruction of atomistic details from coarse-grained structures *J. Comput. Chem.* **31** 1333–43
- [12] Wassenaar T A, Pluhackova K, Böckmann R A, Marrink S J and Tieleman D P 2014 Going backward: A flexible geometric approach to reverse transformation from coarse grained to atomistic models *J. Chem. Theory Comput.* **10** 676–90 PMID: 26580045
- [13] Menichetti R, Kremer K and Bereau T 2018 Efficient potential of mean force calculation from multiscale simulations: solute insertion in a lipid membrane *Biochem. Biophys. Res. Commun.* **498** 282–7
- [14] Hess B, León S, Van Der Vegt N and Kremer K 2006 Long time atomistic polymer trajectories from coarse grained simulations: bisphenol-a polycarbonate *Soft Matter* **2** 409–14
- [15] Zhang G, Chazirakis A, Harmandaris V A and Stuehn T, Daoulas K C and Kremer K 2019 Hierarchical modelling of polystyrene melts: from soft blobs to atomistic resolution *Soft Matter* **15** 289–302
- [16] Brasillo A, Crescitelli S and Milano G 2012 A multiscale approach to triglycerides simulations: from atomistic to coarse-grained models and back *Faraday Discuss.* **158** 479–92
- [17] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A and Bengio Y 2014 Generative adversarial nets *Advances in Neural Information Processing Systems* pp 2672–80
- [18] Arjovsky M, Chintala S and Bottou L 2017 Wasserstein GAN arXiv: [1701.07875](https://arxiv.org/abs/1701.07875)
- [19] Gulrajani I, Ahmed F, Arjovsky M, Dumoulin V, Courville A et al 2017 Improved training of Wasserstein GANs *Advances in Neural Information Processing Systems 30* Guyon I, Luxburg U V, Bengio S, Wallach H, Fergus R, Vishwanathan S and Garnet R eds (Curran Associates, Inc.) pp 5767–5777
- [20] Isola P, Zhu J-Y, Zhou T and Efros A A 2017 Image-to-image translation with conditional adversarial networks *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition* pp 1125–34
- [21] Mirza M and Osindero S 2014 Conditional generative adversarial nets arXiv: [1411.1784](https://arxiv.org/abs/1411.1784)
- [22] van den Oord A, Kalchbrenner N and Kavukcuoglu K 2016 Pixel recurrent neural networks *Proc. of the Conf. on Machine Learning (ICML)* vol **48**
- [23] Gebauer N W, Gastegger M and Schütt K T 2018 Generating equilibrium molecules with deep neural networks arXiv: [1810.11347](https://arxiv.org/abs/1810.11347)
- [24] Noé F, Olsson S, Köhler J and Wu H 2018 Boltzmann generators—sampling equilibrium states of many-body systems with deep learning arXiv: [1812.01729](https://arxiv.org/abs/1812.01729)
- [25] Hoffmann M and Noé F 2019 Generating valid Euclidean distance matrices arXiv: [1910.03131](https://arxiv.org/abs/1910.03131)
- [26] Mansimov E, Mahmood O, Kang S and Cho K 2019 Molecular geometry prediction using a deep generative graph neural network arXiv: [1904.00314](https://arxiv.org/abs/1904.00314)
- [27] Wang W and Gómez-Bombarelli R 2019 Coarse-graining auto-encoders for molecular dynamics *Npj Comput. Mater.* **5** 1–9
- [28] Lamb A, Dumoulin V and Courville A 2016 Discriminative regularization for generative models arXiv: [1602.03220](https://arxiv.org/abs/1602.03220)
- [29] Karras T, Aila T, Laine S and Lehtinen J 2018 Progressive growing of GANs for improved quality, stability, and variation arXiv: [1710.10196](https://arxiv.org/abs/1710.10196)
- [30] Schellenberg J 2009 *Syndiotactic Polystyrene: Synthesis, Characterization, Processing and Applications* (New York: Wiley)
- [31] Fritz D, Harmandaris V A, Kremer K and van Der Vegt N F 2009 Coarse-grained polymer melts based on isolated atomistic chains: Simulation of polystyrene of different tacticities *Macromolecules* **42** 7579–88
- [32] Liu C, Kremer K and Bereau T 2018 Polymorphism of syndiotactic polystyrene crystals from multiscale simulations *Adv. Theory Simul.* **1** 1800024
- [33] Koller D and Friedman N 2009 *Probabilistic Graphical Models: Principles and Techniques* (Cambridge, MA: MIT Press)
- [34] Geman S and Geman D 1984 Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-6** 721–41
- [35] Fukushima K 1980 Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position *Biol. Cybern.* **36** 193–202
- [36] Zhirong W, Song S, Khosla A, Fisher Y, Zhang L, Tang X and Xiao J 2015 3D shapenets: A deep representation for volumetric shapes *2015 Conf. on Computer Vision and Pattern Recognition (CVPR)* pp 1912–20
- [37] Müller-Plathe F 1996 Local structure and dynamics in solvent-swollen polymers *Macromolecules* **29** 4782–91

- [38] He K, Zhang X, Ren S and Sun J 2016 Deep residual learning for image recognition *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition* pp 770–8
- [39] Cohen T S and Welling M 2016 Steerable CNNs arXiv: [1612.08498](https://arxiv.org/abs/1612.08498)
- [40] Xie T and Grossman J C 2018 Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties *Phys. Rev. Lett.* **120** 145301
- [41] Hess B, Kutzner C, Van Der Spoel D and Lindahl E 2008 Gromacs 4: algorithms for highly efficient, load-balanced and scalable molecular simulation *J. Chem. Theory Comput.* **4** 435–47
- [42] Bartók A P, Kondor R and Csányi G 2013 On representing chemical environments *Phys. Rev. B* **87** 184115
- [43] De S, Bartók A P, Csányi G and Ceriotti M 2016 Comparing molecules and solids across structural and alchemical space *Phys. Chem. Chem. Phys.* **18** 13754–69
- [44] Tribello G A, Ceriotti M and Parrinello M 2012 Using sketch-map coordinates to analyze and bias molecular dynamics simulations *Proc. Natl. Acad. Sci.* **109** 5196–201
- [45] Ceriotti M, Tribello G A and Parrinello M 2011 Simplifying the representation of complex free-energy landscapes using sketch-map *Proc. Natl. Acad. Sci.* **108** 13023–8