

Multi-Step Time Series Generator for Molecular Dynamics

Katsuhiko Endo, Katsufumi Tomobe, Kenji Yasuoka

Department of Mechanical Engineering, Keio University
3-14-1 Hiyoshi, Kohoku-ku, Yokohama
Kanagawa, Japan 223-8522

Abstract

Molecular dynamics (MD) is a powerful computational method for simulating molecular behavior. Deep neural networks provide a novel method of generating MD data efficiently, but there is no architecture that mitigates the well-known *exposure bias* accumulated by multi-step generations. In this paper, we propose a multi-step time series generator using a deep neural network based on Wasserstein generative adversarial nets. Instead of sparse real data, our model evolves a latent variable z that is densely distributed in a low-dimensional space. This novel framework successfully mitigates the exposure bias. Moreover, our model can evolve part of the system (*Feature extraction*) with any time step (*Step skip*), which accelerates the efficient generation of MD data. The applicability of this model is evaluated through three different systems: harmonic oscillator, bulk water, and polymer melts. The experimental results demonstrate that our model can generate time series of the MD data with sufficient accuracy to calculate the physical and important dynamical statistics.

1. Introduction

Molecular dynamics (MD) is a standard and powerful tool for investigating molecular behavior, and has been used in various fields, e.g., polymers (Gee, Lacey, and Fried 2006), proteins (Lindorff-Larsen et al. 2011), methane hydrates (Walsh et al. 2009), nucleation (Matsumoto, Saito, and Ohmine 2002), carbon nanotube (Legoas et al. 2003), and metals (Yamakov et al. 2004). The method solves the Newton’s equation of motion and provides the coordinates, velocities, and forces of all the atoms in a system at each time step. Although the applicability of the MD simulations is well known, the high computational costs of large-scale, long-time MD simulations limit the expansion of the method. If we directly calculate two-body forces between all atom pairs, the computational complexity per step is proportional to the square of the number of atoms, which presents a serious bottleneck when the MD simulations are conducted with large number of atoms. Recently, the number of atoms N and time length of the simulation T have reached scales of 10^9 (Shibuta et al. 2017) and 10^{10} (Needham et al. 2016),

respectively; thus, a novel solution to overcome the bottleneck is desired.

Recently, deep neural networks have achieved significant results in many fields. Recurrent neural networks (RNNs) (Elman 1990), long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997), and other derivative models (Mogren 2016; Fabius and van Amersfoort 2014) have led to remarkable advances in both discriminative and generative sequence data tasks. However, when the generative models are used repeatedly, the generations are affected by previous ones, leading to an accumulation of bias, called the *exposure bias*. In a naive generator, this bias causes the probability of error to grow quadratically with the iteration number (Ross and Bagnell 2010). Various studies have attempted to mitigate the exposure bias. In the natural language processing field, many studies use the beam search to mitigate effects of the exposure bias (Sutskever, Vinyals, and Le 2014). Some studies have mitigated the bias by using the specific loss functions that directly evaluate the correlations with the training sequence (Shen et al. 2015). A recent study by Yu et al. used extended generative adversarial nets (GANs) (Yu et al. 2017). Although the above studies achieved some success, our task of generating the MD data suffers from more serious exposure bias. The computational costs of the MD simulations mean that we would like to obtain long-term dynamical statistics from short-time-length MD data. To fulfill this requirement, we have to generate the short-time-length MD data repeatedly and concatenate them to make a long-time-length MD data; therefore, a novel framework for mitigating the exposure bias is needed.

To develop a new deep neural generative model, we introduce a multi-step time series generator (MD-GAN) that efficiently samples time series of the MD simulations (see Fig. 1A). Our model has the following advantages:

1. **Mitigation of the exposure bias:** as we consider time series generation not as $p(X_t|X_{t-1})$ but as $p(X_{t-1}|z_{t-1}), p(X_t|z_t)$, and $p(z_t|z_{t-1})$, our model successfully mitigates the exposure bias.
2. **Feature extraction (contributing to the efficient sampling):** our model can simulate the time evolution of a part of the whole system.
3. **Step skip (contributing to the efficient sampling):** it can operate with any time step; thus, users can set a suitable

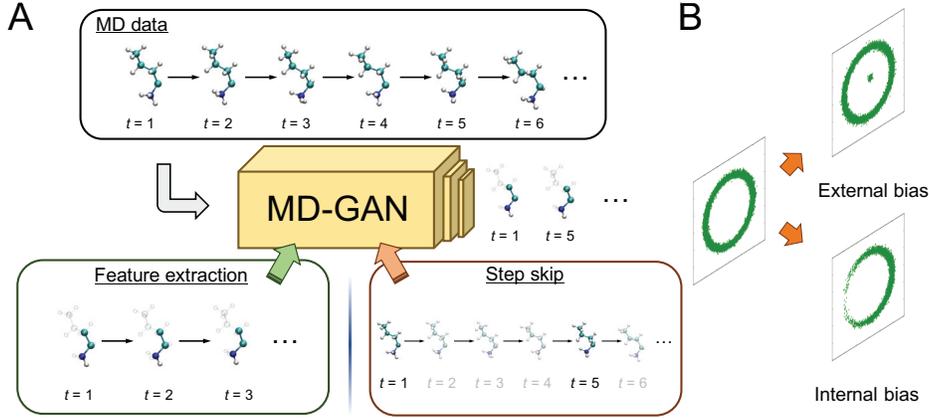


Figure 1: Schematic view of our concepts. (A) The proposed multi-step time series generator for MD. Using the advantages of *feature extraction* and *step skip*, our model efficiently generates MD time series data. (B) Schematic view of the internal and external biases. In the external bias, data evolve outside of the manifold. All data are distributed in the manifold; however, their density and distribution changes in the case of internal bias.

time step for the target dynamics.

4. **Probabilistic time evolution:** thanks to the Wasserstein GANs (Arjovsky, Chintala, and Bottou 2017), our model gives a probabilistic time evolution.
5. **Identification of latent structures:** our model reveals the latent structure of the data through the behavior of the latent variable.

2. Preliminaries

Let us first consider a deterministic system and the evolution equation

$$\frac{d}{dt}\mathbf{x}(t) = f(\mathbf{x}(t)),$$

where f is the function for time evolution determined by scientific considerations and $\mathbf{x}(t) \in \mathbb{R}^D$ is state variables in a D -dimensional space at time t . From this equation, we obtain the time series from the initial state $\mathbf{x}(0)$. In computer simulations, the equation f adopts a finite-difference approximation in both the spatial and temporal dimensions to give a discretized equation $f_{\Delta t}$ with a time step Δt . The time step Δt depends on the required precision, e.g., femtosecond timescale is required in all-atom MD simulations. The discretized equation $f_{\Delta t}$ evolves the state of the system as $\mathbf{x}_{n+1} = f_{\Delta t}(\mathbf{x}_n)$ where \mathbf{x}_n is $\mathbf{x}(n\Delta t)$.

Consider an extracted feature $\mathbf{y}_n \in \mathbb{R}^{D'}$ ($D' \leq D$) which contains rich physical information but exists in a lower-dimensional space (see Fig. 1 *Feature extraction*). However, unlike \mathbf{x}_n , deterministic equations for most \mathbf{y}_n are unknown. Instead of the deterministic evolution, we consider stochastic evolution represented by a conditional probability of \mathbf{y}_n given all the previous states $p(\mathbf{y}_n | \mathbf{Y}_{n-1:0})$, where $\mathbf{Y}_{b:a:s}$ is the time series of features with time step s from time a to time b ($\mathbf{y}_b, \mathbf{y}_{b-s}, \dots, \mathbf{y}_a$). By convention, s is omitted when $s = 1$. When calculating dynamical statistics that only depends on the time series of the features \mathbf{Y} , we calculate

the statistics from the time series obtained by the stochastic evolution $p(\mathbf{y}_n | \mathbf{Y}_{n-1:0})$. Moreover, when the dynamical statistics of interest do not depend on small-step dynamics, we use only the step-skipped stochastic evolution $p(\mathbf{y}_{Kn} | \mathbf{Y}_{Kn-1:0:K})$ (see Fig. 1 *Step skip*).

Our goal is to obtain the stochastic evolution $p(\mathbf{y}_n | \mathbf{Y}_{n-1:0})$. Unfortunately, this probability distribution is usually intractable. Machine learning can be applied to obtain an approximated distribution for the stochastic evolution. In this paper, we assume the following two assumptions:

- **Assumption 1:** the stochastic evolution of \mathbf{y}_i does not depend on \mathbf{y}_{i-m} ($m > M$), $p(\mathbf{y}_n | \mathbf{Y}_{n-1:0}) = p(\mathbf{y}_n | \mathbf{Y}_{n-1:n-M})$.
- **Assumption 2:** the distribution of the features \mathbf{y}_n is stationary and the stochastic evolution is time invariant. In this model, we focus on the physical “equilibrated systems”, which means that $p(\mathbf{y}_n)$ is stationary, and we can use the same distribution for stochastic evolution at any time.

With these assumptions, we construct a novel generative model for time series. Let us consider $p(\mathbf{Y}_{(k+1)M-1:kM} | \mathbf{Y}_{kM-1:0})$. According to assumption 1, this can be denoted by $p(\mathbf{Y}_{(k+1)M-1:kM} | \mathbf{Y}_{kM-1:(k-1)M})$. Using this sequence-to-sequence probability distribution, we form the time series as a generative model; however, when we use the generative model iteratively, we encounter the *exposure bias* (Ross and Bagnell 2010). In multi-step generations, the generative model uses the previous output as the real input, and this includes some small bias from the incompleteness of the model. Over the multi-step generations, this small bias accumulates and becomes non-negligible.

To clarify the problem, we assume that this bias can be divided into two types: *internal* and *external* biases. In many

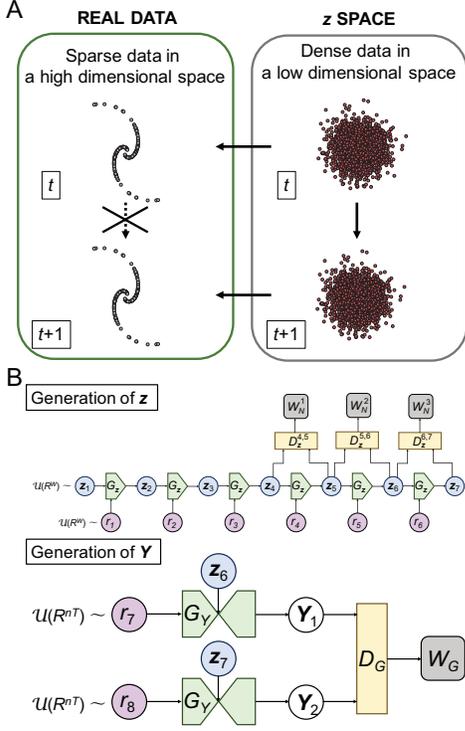


Figure 2: Architecture of our model. (A) Effects of the latent variable z . Instead of the sparse real data distributed in a high-dimensional space, we evolve the dense latent variable z , which are embedded in a low-dimensional space. (B) Architecture of our model. G , D , W , Y , z , and r_k represent a generator, a discriminator, the Wasserstein distance, features, a latent variable, a random vector, respectively. After determining the latent variable (top), D_G discriminates Y_1 and Y_2 which are generated using the latent variable (bottom).

cases, y_n lies along a low-dimensional manifold embedded in a high-dimensional space (Basri and Jacobs 2016; Lee et al. 2003; Basri and Jacobs 2003). We define external bias as that when the predicted data are located outside this manifold; in contrast, internal bias is defined as the bias located in the manifold but different from the real data (see Fig. 1B). To mitigate both the biases, we propose the following latent variable model. In this model, the probability distribution of the features $p(\mathbf{Y}_{(k+1)M-1:kM})$ depends on the latent variable z_k as follows:

$$p(\mathbf{Y}_{(k+1)M-1:kM}) = \int p(\mathbf{Y}_{(k+1)M-1:kM} | z_k) p(z_k) dz_k.$$

z_k contains all the information of $\mathbf{Y}_{(k+1)M-1:kM}$ and is densely distributed in a low-dimensional space (see Fig. 2A). The model generates time series of the features using $p(\mathbf{Y}_{(k+1)M-1:kM} | z_k) p(z_k)$. Let us set z_k as being embedded in a low-dimensional space. As we have constructed dense variables z_k in a low-dimensional space, the out-of-manifold probability decreases, which mitigates the external bias. In the internal case, assumption 2 contributes to

the mitigation of the bias. The distribution of feature y_t is stationary; thus, the internal bias becomes small under the constraint $p(z_k) = p(z_{k-1})$.

3. Proposed architecture

Architecture of our model

In this section, we implement the novel generative model described in Section 2. The Wasserstein GAN (WGAN) (Arjovsky, Chintala, and Bottou 2017) is used to approximate the probability distribution. The WGAN has two networks: a generator $G : z \rightarrow Y$ and a discriminator $D : Y \rightarrow \mathbb{R}$. G generates Y from the probability distribution p_G with a seed $z \sim p(z)$, and D calculates the Wasserstein distance (WD) between the generated distribution p_G and the objective distribution p_R defined as

$$W(p_G, p_R) = \sup_{\|g\|_L \leq 1} \mathbb{E}_{x \sim p_G}[g(x)] - \mathbb{E}_{x \sim p_R}[g(x)], \quad (1)$$

where the supremum is over all the 1-Lipschitz functions g . The network G is trained to minimize this WD, and the generator captures the objective distribution after the training.

Figure 2B shows the architecture of the proposed model. Visual icons show multi-layer affine networks (green pentagon), U-net based networks with z inserted in the bottleneck of the U-net (green bow tie), a random vector (pink circle), the latent vector (blue circle), a discriminator (yellow rectangle), and the generated data (white circle).

Let $z_1 \sim \mathcal{U}(R^W)$ be sampled from W dimensional uniform distribution between 0 and 1. G_z is a generator and generates z using previous z and a random value sampled from $\mathcal{U}(R^W)$. Then, we apply G_z to z_1 , and it generates z_2, z_3, \dots , repeatedly. D_z is a discriminator and estimates the WD between z_k and z_{k+1} to make the distribution $p(z)$ stationary. Since we consider that the former z (z_1, z_2, z_3 , and z_4) are not stationary, we only apply the D_z to the later pairs of z . After a consecutive pair of z is randomly picked up (z_5 and z_6 , or z_6 and z_7 , or \dots), G_y generates Y_1 and Y_2 using the pair of z . D_g is a discriminator and discriminates the generated Y_1 and Y_2 and real data. Training process of this network is as follows: (1) G_z is trained to minimize the total WD estimated by all D_z and D_g , (2) G_y is trained to minimize the WD estimated by D_g , (3) Each D_g was trained to estimate the WD between the pair of z , (4) D_z is trained to estimate the WD between the generated and real data. At the inference, time series z is generated by G_z from $z_1 \sim \mathcal{U}(R^W)$, the former some steps is excluded, and time series Y is generated from z_k by G_Y .

Inside structure of the networks

The G_Y consists of the U-net (Ronneberger, Fischer, and Brox 2015) based networks, which transform uncorrelated multi dimensional time length M normal distributions into some distributions. The D_Y is multi-layer convolutional network. In our experiments, the U-net based networks achieved more accurate than the network only having deconvolution. We found that deconvolution in the G_Y achieved most accurate when the stride size was equal to kernel size.

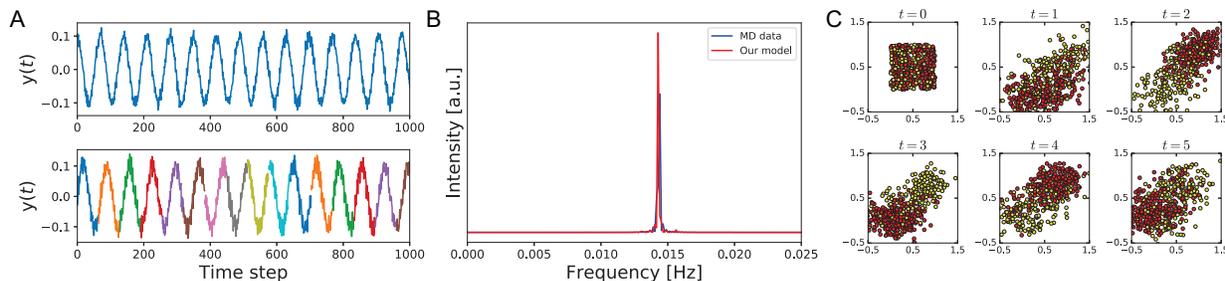


Figure 3: Motion of harmonic oscillator with noise. (A) Time series of the reference (upper) and that generated by our model (bottom). In the generated time series, each color represents the sequence data generated in one generation. (B) Spectra of the reference and generated time series. (C) Behavior of the latent variable in our model. The x and y axes show third and fourth dimension of the latent variable. Yellow/red circles represent second dimension of the latent variable greater than/less than 0.5 at $t = 3$, respectively. The results show that the latent variable obtain a periodic structure.

The number of convolution layers is same as that of deconvolution layers as well as the U-net. The G_z and D_z consist of multi-layer affine networks. Training of G_z can be dramatically fast when the Residual Network (ResNet) was used (He et al. 2016). Other parameters of the networks are needed to be tuned for each dataset.

Advantages of the proposed model for MD

This model has a number of advantages for the rapid prediction of dynamical statistics. The mitigation of the exposure bias is one of the novel advantages for the MD simulations. Because of the dense variables z embedded in a low-dimensional space, we can obtain long-term dynamical statistics using multi-step time evolutions. The other advantage is that this model needs only $2M$ steps of real data for the training. In the MD simulations, we have to calculate time series of the state variables x that are much longer than the time length of the dynamics of interest. In many cases, this is the bottleneck in the scaling of MD simulations. Therefore, the short real time series requirement enhances the computational efficiency of the simulations. In addition to the above advantages, as our model uses the latent variable z in a low-dimensional space, it is expected that the structures of the time series features y_n are mapped into $p(z)$. Thus, we are able to extract hidden time-directional structures of the features. Moreover, compared with LSTM, our model is easier to analyze behavior of the latent variable because our model has only one latent variable. This characteristic provides a great benefit for the MD studies through this high interpretability.

4. Experimental results

We performed three experiments: harmonic oscillator with noise, water vibrational spectra, and polymer dynamics in melts. Each experiment was performed by following four steps: (1) running short and long MD simulations to get time series of x for learning and comparison, respectively, (2) getting the pair of Y (the consecutive pair of 64 steps of the extracted features) by *Feature extraction* and *Step skip*,

Table 1: Dimensions of the parameters in the experiments.

Experiment	Harmonic	Water	Polymer
x	1	2x3x330	2x3x30000
Y	64x1	64x3	64x3
z	16	16	16

(3) training the model using Y , and (4) comparing the generated and real long data. Table 1 shows dimensions of the parameters (x , Y , and z) in the experiments.

Harmonic oscillator with noise

We first demonstrate that our model can evolve features while mitigating the exposure bias. Figure 3A shows our results for a harmonic oscillator with noise, where the top and bottom figures represent the reference and generated time series, respectively. The reference time series was generated by the equation

$$0.1\sin(0.7t) + \eta,$$

where $\eta \sim \mathcal{N}(0, 0.01)$. Our model provides time series containing 64 steps, and each color shows the time series of each generation in Figure 3A (bottom). This result clearly shows that our model can generate time series with the correct amplitude, frequency, and noise. Remarkably, these quantities did not change after multi-step generations, indicating that our model successfully mitigates the exposure bias. Moreover, when the frequency changes due to the bias, the generated time series exhibits self-modification. Figure 3B shows the vibrational spectra obtained by the Fourier transform of the time series. The frequency and intensity of the mode in the generated time series are in good agreement with those in the reference. Thus, the generated time series quantitatively match the reference. To compare with other machine learning models, we calculated mean square error of the normalized spectra with real data. When we extended 128 into 1024 steps, the performance of our model is 79 and 104 times better than that of a naive model of the proposed architecture and C-RNN-GAN (Mogren 2016). The naive model

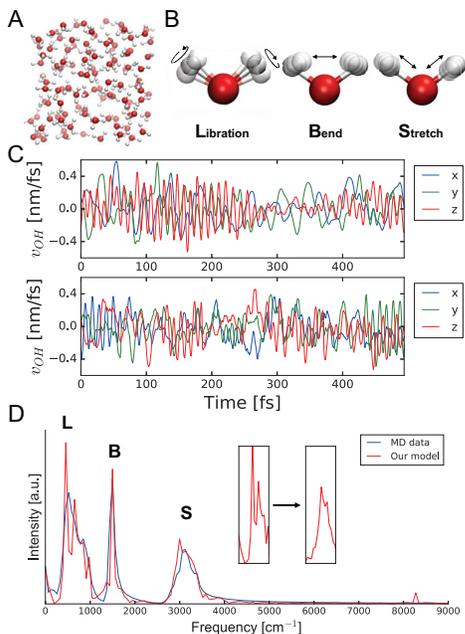


Figure 4: OH vibrational spectra of water molecules in the equilibrium bulk system. (A) Snapshot of the system containing 110 water molecules. (B) Schematic view of vibrational motion of the water molecule. (C) OH relative velocities of the MD (top) and generated data by our model (bottom). Blue, green, and red lines represent x , y , and z components of the velocity vector, respectively. (D) OH vibrational spectra of water molecules using the MD and generated time series. The spectra were obtained by the Fourier transform of the OH relative velocity correlation function. Insets show that larger training data reduce the noise of spectrum.

has a generator without latent variables and a discriminator. In this experiment, the dimensions of the latent variable was set to be 16. Let us investigate the behavior of the latent variable. Figure 3C shows transitions of the latent variable in the generation of z after training. The x and y axes show third and fourth dimension of the latent variable, and the yellow and red denote whether the second dimension of the latent variable was greater or less than 0.5 at $t = 3$, respectively. At $t = 0$, all the latent variable were generated from the uniform distribution. After the initialization, the distribution of z reached a stationary distribution through our generator G_z . We can identify a periodic structure in the stationary distribution of z , and this is one of the example where the latent variable successfully extract the interesting structures in the target system. This advantage of our model will contribute to an understandings of the system behavior.

Water vibrational spectra

Next, we applied our model to a more complex system: the bulk water. Water molecules are an important target of research in chemical, physical, and biological fields. Because of thermodynamic fluctuations and interactions with

other molecules, the OH bonds of water molecules have specific vibrational motions. We performed *ab initio* MD simulations, which is an MD method on the basis of *ab initio* energy calculations, for the bulk water (see Fig. 4A). To test the applicability for real time series, we used the OH motions as an input and calculated vibrational spectra using the generated time series. For the *ab initio* MD simulations, we used the CPMD code (Hutter and Iannuzzi 2005) using the Car-Parrinello method (Car and Parrinello 1985). In the CPMD simulations, the Perdew-Burke-Ernzerhof functional was used to approximate the exchange-correlation terms (Perdew, Burke, and Ernzerhof 1996), and we described the valence-core interaction using the Martins-Troullier pseudo-potentials (Troullier and Martins 1991). The CPMD simulation was conducted under the isothermal and iso-volume conditions (305 K), and the volume of the system was predefined by the classical isothermal and isobaric classical MD simulation using the NAMD 2.9 software (Phillips et al. 2005) with the TIP3P water model (Jorgensen et al. 1983). In this experiment, the OH relative velocities were prepared as input for our model, and our model generated time series of OH velocities. A notable point is that input velocities are not long time enough to calculate the statistic ($M = 64$ steps = 61.92 fs). Figure 4C shows the OH velocities of the MD and generated data in the water molecules. As this figures shows, the vibrational motion of the water molecules is complex. Vibrational spectra were obtained by taking the Fourier transform of the OH relative velocity correlation function (the vibrational density of states) (Tanzi, Ramondo, and Guidoni 2012),

$$P(\omega) = \sum_{k=1}^N \int_{-\infty}^{\infty} \langle \dot{\mathbf{r}}_k(0) \dot{\mathbf{r}}_k(t) \rangle e^{i\omega t} dt,$$

where $\dot{\mathbf{r}}_i(0) \dot{\mathbf{r}}_i(t)$ is the time series velocity auto-correlation function and $\langle \rangle$ is the ensemble average. Figure 4D shows OH vibrational spectra from the MD and generated time series of OH motion in water molecules. The OH vibrational motion is characterized by three modes: libration (600 cm^{-1}), bend (1500 cm^{-1}), and stretch (3200 cm^{-1}) motions (see Fig. 4B). In the spectra given by CPMD, we identified three peaks corresponding to these modes. Notably, the generated motion also contained this information, and there are three clear peaks with precise frequencies. Furthermore, the external forms of the frequency distribution, including slopes and intensities, matched each other. Real vibrational motion includes anharmonic combinations of the peaks, and interactions with external environments make the spectra significantly complex. Under this environment, our model precisely generated the time series of OH bonds, and we could obtain physical and important dynamical statistics using the generated time series. Unfortunately, there is one unexpected peak at around 8300 cm^{-1} . This peak is unphysical and can be attributed to the bias. Note that larger training data reduce a noise of the libration peak (see Fig. 4D insets).

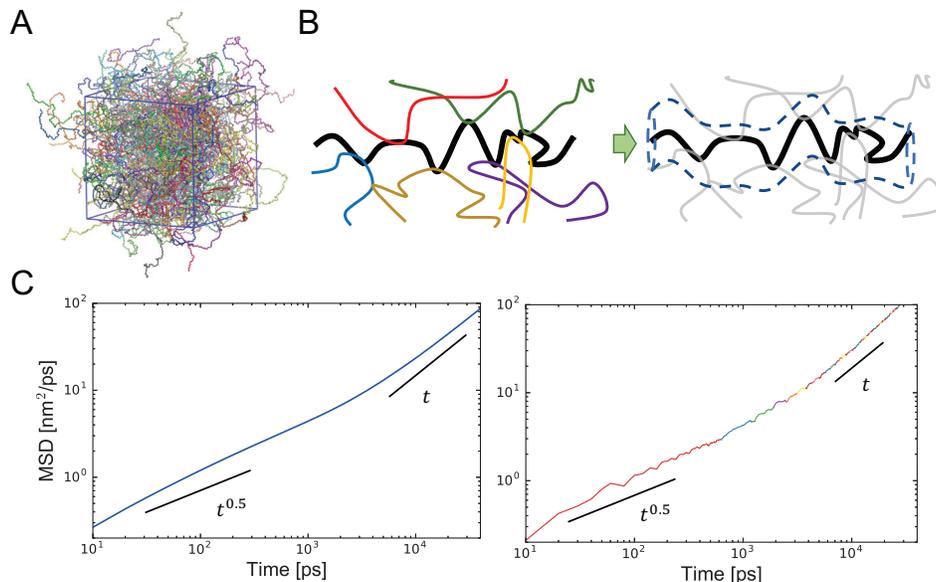


Figure 5: MD simulations of polymer PE melts. (A) Snapshot of the system containing 300 PE polymers. The molecular weight is 1405 g/mol. (B) Schematic representation of entangled polymer melts and snake-like motion in a tube. Until the end-to-end relaxation time, the polymers end-to-end relaxation time scales with polymer length. (C) Mean square displacement of the chain center using (left) the MD data and (right) generated data by our model. In the generated data, each color represents the sequence data generated in one generation.

Polymer dynamics in melts

Polymer materials have unique and complex dynamics in the melts, and the dynamics depends on various molecular and thermodynamic parameters. We performed the MD simulations of polymer polyethylene (PE) melts to simulate the complex dynamics of the polymer. The MD simulations were conducted using the TraPPE-UA force field (Martin and Siepmann 1998) and the GROMACS package (Pronk et al. 2013) under the same conditions and preparations as a previous study (Takahashi et al. 2017). Figure 5A shows a snapshot of the system filled with 300 PE polymer chains. In the MD simulations, the diffusion coefficient can be characterized by the mean square displacement (MSD):

$$g(t) = \langle [\mathbf{r}(t) - \mathbf{r}(0)]^2 \rangle,$$

where $\mathbf{r}(t)$ is the coordinate vector at time t . It is known that the slope of the MSD (diffusion coefficient) of the chain center anomaly changes with time because of the chain entanglement (Takahashi et al. 2017). Figure 5B shows a schematic view of the chain entanglement and the concept of tube theory (Doi and Edwards 1978). In tube theory, chain entanglement is described as a tube constraint, and the polymer chain can only move along the major axis like a snake (reptation motion). Figure 5C shows the MSDs of the chain center of the polymers using time series generated by the MD simulation and our model. In our model, each color shows each generation which contains 64 steps (= 640 ps). In the MD simulation results, the slopes of the MSDs (i.e., diffusion coefficient) changes from $t^{<0.5}$ to t^1 ; this transition time is called the end-to-end relaxation time τ_R . This

retardation can be described as the tube constraint by the entanglement of surrounding polymers, and τ_R corresponds to the chain length of the polymer (see Fig. 5B right). In the result given by our model, the MSD also has a transition time τ_R . Remarkably, τ_R is much longer than the length of the one-step sequence; thus, our model generates sequence data from the precise phase space (a space containing all possible states of the system), and the generated time series enclose the short- and long-term dynamics of the polymers.

Table 2 shows calculation performance of the MD simulation and our model with one process of the Intel Core i7-4930K. Using one of the fastest MD package GROMACS, whole calculations reached 55.2 days when we obtained the averaged MSD of the polymers for 40 ns. In our model, total calculation time was 67.9 hours in this experiment, and most calculation time was the training. As a result, we achieved our goal of obtaining the averaged MSD of chain center 19 faster than the MD simulations. These results show that the time evolutions given by our model are precise enough to simulate the complex dynamics of the polymers; moreover, our model much more efficiently generates time series of the target features than the MD simulations.

4. Conclusions and future work

In this paper, we have presented a multi-step time series generator using the Wasserstein GAN-based deep neural network to efficiently generate time series of MD simulations. Using our novel framework, the model successfully mitigates the exposure bias. MD simulation research will ben-

Table 2: Calculation performance of the MD simulation and our model with one process of the Intel Core i7-4930K.

	MD (GROMACS)	Our model
Speed	1.38 day/ns	67.9 h/model
Speed (/40 ns MSD)	55.2 days	2.83 days

efit from several advantages (e.g., *step skip*, *feature extraction*, and the structures of the latent variable z). Using various systems (harmonic oscillator, bulk water, and polymer melts), we have demonstrated the capability of our model to generate time series that are sufficiently precise to calculate physical and important dynamical statistics. However, it remains difficult to verify that the assumption 1 was satisfied and the trained probability was the same as that of the real data. In future work, we will study five aspects. (1) The latent variable z can extract the important structures of the system in a low-dimensional space; however, it is still difficult to analyze the characteristics of the structures. Some suitable analysis methods are required. (2) In our model, there are two generators, G_z and G_Y ; thus, it is difficult to determine when to terminate the learning step, because the convergence properties of the two generators are basically different. Other multi-network coexistence architectures (e.g., GAN) also suffer from this drawback. We will attempt to provide some metric for finding the best point between over- and under-fitting. (3) The model requires sensitive adjustment of hyperparameters, as in other deep neural networks. (4) Our results showed that dynamics properties (e.g., MSD) can be calculated using smaller time length than the time length of the dynamics properties. However, it is unclear what time length is required. (5) Finally, there is no method to verify that the generated data follows the probability distribution of the real data. Future works will also include an improved model that is easier to use.

Acknowledgments

This work is supported in part by Ministry of Education, Culture, Sports, Science and Technology (MEXT) as Research and Development of Next-Generation Filed. K.T. is supported by MEXT Grant-in-Aid for the Program for Leading Graduate Schools.

References

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875*.

Basri, R., and Jacobs, D. W. 2003. Lambertian reflectance and linear subspaces. *IEEE transactions on pattern analysis and machine intelligence* 25(2):218–233.

Basri, R., and Jacobs, D. 2016. Efficient representation of low-dimensional manifolds using deep networks. *arXiv preprint arXiv:1602.04723*.

Car, R., and Parrinello, M. 1985. Unified approach for molecular dynamics and density-functional theory. *Phys. Rev. Lett.* 55(22):2471.

Doi, M., and Edwards, S. 1978. Dynamics of concentrated polymer systems. Part 1.-Brownian motion in the equilibrium state. *Journal of the Chemical Society, Faraday Transactions 2: Molecular and Chemical Physics* 74:1789–1801.

Elman, J. L. 1990. Finding structure in time. *Cognitive science* 14(2):179–211.

Fabius, O., and van Amersfoort, J. R. 2014. Variational recurrent auto-encoders. *arXiv preprint arXiv:1412.6581*.

Gee, R. H.; Lacevic, N.; and Fried, L. E. 2006. Atomistic simulations of spinodal phase separation preceding polymer crystallization. *Nat. Mater.* 5(1):39–43.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Comput.* 9(8):1735–1780.

Hutter, J., and Iannuzzi, M. 2005. CPMD: Car-Parrinello molecular dynamics. *Zeitschrift für Kristallographie-Crystalline Materials* 220(5/6):549–551.

Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; and Klein, M. L. 1983. Comparison of simple potential functions for simulating liquid water. *J. Chem. Phys.* 79(2):926–935.

Lee, K.-C.; Ho, J.; Yang, M.-H.; and Kriegman, D. 2003. Video-based face recognition using probabilistic appearance manifolds. In *Computer vision and pattern recognition, 2003. proceedings. 2003 ieee computer society conference on*, volume 1, I–I. IEEE.

Legoas, S.; Coluci, V.; Braga, S.; Coura, P.; Dantas, S.; and Galvao, D. 2003. Molecular-dynamics simulations of carbon nanotubes as gigahertz oscillators. *Phys. Rev. Lett.* 90(5):055504.

Lindorff-Larsen, K.; Piana, S.; Dror, R. O.; and Shaw, D. E. 2011. How fast-folding proteins fold. *Science* 334(6055):517–520.

Martin, M. G., and Siepmann, J. I. 1998. Transferable potentials for phase equilibria. 1. United-atom description of n-alkanes. *J. Phys. Chem. B* 102(14):2569–2577.

Matsumoto, M.; Saito, S.; and Ohmine, I. 2002. Molecular dynamics simulation of the ice nucleation and growth process leading to water freezing. *Nature* 416(6879):409–413.

Mogren, O. 2016. C-RNN-GAN: Continuous recurrent neural networks with adversarial training. *arXiv preprint arXiv:1611.09904*.

Needham, S. R.; Roberts, S. K.; Arkhipov, A.; Mysore, V. P.; Tynan, C. J.; Zanetti-Domingues, L. C.; Kim, E. T.; Losasso, V.; Korovesis, D.; Hirsch, M.; et al. 2016. EGFR oligomerization organizes kinase-active dimers into competent signalling platforms. *Nat. Commun.* 7:13307.

Perdew, J. P.; Burke, K.; and Ernzerhof, M. 1996. Generalized gradient approximation made simple. *Phys. Rev. Lett.* 77(18):3865.

Phillips, J. C.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R. D.; Kale, L.; and

- Schulten, K. 2005. Scalable molecular dynamics with NAMD. *J. Comput. Chem.* 26(16):1781–1802.
- Pronk, S.; Páll, S.; Schulz, R.; Larsson, P.; Bjelkmar, P.; Apostolov, R.; Shirts, M. R.; Smith, J. C.; Kasson, P. M.; van der Spoel, D.; et al. 2013. GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit. *Bioinformatics* 29(7):845–854.
- Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 234–241. Springer.
- Ross, S., and Bagnell, D. 2010. Efficient reductions for imitation learning. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 661–668.
- Shen, S.; Cheng, Y.; He, Z.; He, W.; Wu, H.; Sun, M.; and Liu, Y. 2015. Minimum risk training for neural machine translation. *arXiv preprint arXiv:1512.02433*.
- Shibuta, Y.; Sakane, S.; Miyoshi, E.; Okita, S.; Takaki, T.; and Ohno, M. 2017. Heterogeneity in homogeneous nucleation from billion-atom molecular dynamics simulation of solidification of pure metal. *Nat. Commun.* 8.
- Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, 3104–3112.
- Takahashi, K. Z.; Nishimura, R.; Yasuoka, K.; and Masubuchi, Y. 2017. Molecular dynamics simulations for resolving scaling laws of polyethylene melts. *Polymers* 9(1):24.
- Tanzi, L.; Ramondo, F.; and Guidoni, L. 2012. Vibrational spectra of water solutions of azoles from QM/MM calculations: Effects of solvation. *J. Phys. Chem. A* 116(41):10160–10171.
- Troullier, N., and Martins, J. L. 1991. Efficient pseudopotentials for plane-wave calculations. *Phys. Rev. B* 43(3):1993.
- Walsh, M. R.; Koh, C. A.; Sloan, E. D.; Sum, A. K.; and Wu, D. T. 2009. Microsecond simulations of spontaneous methane hydrate nucleation and growth. *Science* 326(5956):1095–1098.
- Yamakov, V.; Wolf, D.; Phillpot, S.; Mukherjee, A.; and Gleiter, H. 2004. Deformation-mechanism map for nanocrystalline metals by molecular-dynamics simulation. *Nat. Mater.* 3(1):43.
- Yu, L.; Zhang, W.; Wang, J.; and Yu, Y. 2017. SeqGAN: Sequence generative adversarial nets with policy gradient. In *AAAI*, 2852–2858.