Denoising diffusion probabilistic models for replica exchange

Yihang Wang† and Pratyush Tiwary*,‡

†Biophysics Program and Institute for Physical Science and Technology, University of Maryland, College Park 20742, USA.
‡Department of Chemistry and Biochemistry and Institute for Physical Science and Technology, University of Maryland, College Park 20742, USA.

E-mail: ptiwary@umd.edu

Abstract

Using sampling at higher temperatures to learn about the energy landscape at a lower temperature is perhaps the most na¨ıve approach one can think of to solve the rare event problem in molecular dynamics. However, this is hard to do in practice. Arguably the only reliable scheme that exploits the ease of travelling across the energy landscape at higher temperatures is the replica exchange molecular dynamics (REMD) scheme. But does REMD make maximal use of the sampling across temperatures, or could we do better? Here we demonstrate a method that by post-processing a REMD run can directly make use of the sampling at all temperatures at the same time, outperforming the direct estimates from REMD. At the heart of our method lies the use of denoising diffusion probabilistic models (DDPM), which are a class of generative artificial intelligence. This allows us to directly sample from the joint probability distribution in configuration and temperature space, wherein the key idea is to treat the temperature as a random variable and not a control parameter as usually done in REMD. We demonstrate this method’s power to (i) generate samples from metastable states and transition states that the lowest replica in REMD never even visited, (ii) provide more reliable free energy estimates across configuration space and (iii) interpolate, and to some extent also extrapolate across temperatures for replicas that were never simulated. The results here are demonstrated for REMD performed on a 9-residue peptide undergoing left- to right-handed helix transitions in water, where REMD is performed by exchanging all 4749 protein and water atoms across different replicas.

The temperature $T$ is traditionally considered a control parameter when studying the equilibrium states of $N$−body systems. Given $T$, or equivalently the inverse temperature $\beta = 1/k_B T$ (where $k_B$ is Boltzmann’s constant), molecular dynamics (MD) or Monte Carlo (MC) methods allow sampling configurations $x$ as per the equilibrium probability $p_\beta(x) \equiv e^{-\beta U(x)}/Z$, where $U(x)$ is the potential energy of the $N$−body system and $Z = \int dx e^{-\beta U(x)}$ is the partition function. For systems of practical interest in biology, chemistry and materials science, if $T$ is not large enough, it becomes nearly impossible to sample reliably from $p_\beta(x)$ as many regions of interest in configuration space will have $e^{-\beta U(x)} \approx 0$. One popular workaround for this limitation is the Replica exchange molecular dynamics (REMD)¹² method, which can be used without any prior knowledge of the system’s dominant slow degrees of freedom. In REMD one simulates $K+1$ replicas of the system at temperatures $\beta = \beta_0 > \beta_1 > ... > \beta_K$. For low enough $\beta_K$ or equivalently high enough temperature $T_K$, the sampling from
\[ p_{\beta_K}(x) \equiv e^{-\beta_K U(x)} / Z_K \] is expected to be more ergodic. One then periodically exchanges configurations between consecutive pairs of replica with a Metropolis-type acceptance probability that depends on the potential energies of the two replicas and their temperatures. This way even the low-temperature \( \beta_0 \) replica can explore configurations that it would not have otherwise visited. The development of computational resources, especially the support of massively parallel computing, has made REMD an accessible and reliable approach to sample rare events in complicated systems. This simple procedure has been extremely powerful over the decades for the study of molecular systems with rough energy landscapes for fundamental science and practical applications. Numerous advances have been introduced over this basic idea in order to make it more efficient computationally and it continues to be an area of very active research.

In this letter we take a slightly different view of the temperature \( \beta \) in REMD and consider it as a random variable instead of a control parameter. Our motivation is that all replicas across different temperatures can be viewed as being sampled from the same joint probability \( p(x, \beta) \) – as opposed to different replicas sampling from respective \( p_{\beta_k}(x) \). This is feasible for systems studied in MD simulation where \( N \) is much smaller than the thermodynamic limit, and thus the instantaneous temperature \( T \) for each replica, derived from its instantaneous kinetic energy, naturally shows significant fluctuations relative to its mean. Indeed, these fluctuations are the key to a non-zero acceptance probability across replica. Thus, perhaps by treating temperature just as a control parameter REMD is not making full use of the information gathered across the ladder of temperatures. In arguably most current incarnations of REMD (excluding exceptions such as Ref. [13]), all that the higher temperatures do is to help the lower temperature replicas discontinuously appear in different locations of the configuration space. Here we demonstrate a pathway to sampling from the full joint probability \( p(x, \beta) \) in a way that can complement REMD. To do so we work with the instantaneous temperature of the system associated with its kinetic energy instead of the temperature of the heat bath that we expect the thermostat to enforce on average. More rigorously thus, we are sampling the joint \( p(x, \kappa) \) where the per-particle kinetic energy \( \kappa \) is related through its ensemble average to the temperature, i.e. \( \langle \kappa \rangle = \frac{3}{2} T \). We demonstrate how such a post-processing procedure significantly improves the estimates of free energies made through REMD and providing accurate sampling in parts of configuration that were not previously visited in the lowest temperature replica during REMD. This could be metastable states or transition states. We also show how this can be used to generate samples at temperatures not included in the ladder of replicas.

Our task at hand now is to learn the joint probability \( p(x, \beta) \) given sampling that has already been performed in REMD across \( x \)-space and temperatures \( \beta = \beta_0 > \beta_1 > ... > \beta_K \). There are two main challenges in this. The first challenge comes from the curse of dimensionality: the memory or computational resources needed to track a very high-dimensional distribution function increase exponentially as the number of degrees of freedom increases. For REMD of a small 9-residue peptide in explicit water, which we show in this letter, we exchange all 4,749 atomic coordinates, but for the purpose of analysis we set \( x \) as 18 Ramachandran dihedral angles. This means we already have a 19-dimensional space where binning procedures are out of the question. The second challenge comes from the sparsity of the data. Most of our samples come from high probability regions \( p(x, \beta) \) and we have very few samples for low-probability states. In summary we have sparse sampling of data points in very high-dimensional \((x, \beta)\)-space and wish to construct \( p(x, \beta) \) from this information.

To learn a \( p(x, \beta) \) from such a high-dimensional and sparse dataset, we use the denoising diffusion probabilistic model (DDPM) [17] as a generative artificial intelligence (AI) model inspired by non-equilibrium dynamics to approximate and sample from very high-dimensional distributions. DDPM has been shown to possess the ability to infer and learn
Figure 1: Lower panel shows the neural network architecture used in this work. It has the basic structure of the U-Net model. During the diffusion process, which is indicated by the direction of blue arrows in the upper panel, noise is gradually added to the sampled data, in this case the picture of a very good boy, through a diffusion process labeled with diffusion step $t_i$. This changes the sampled distribution (for example, more pictures of dogs) to a simpler isotropic Gaussian distribution from which one can easily generate more samples. An AI model is then trained to reverse such a diffusion process and starting from sampled noise, learn to generate images similar to the input image by following the direction indicated by the orange arrows. It takes an 1-d array $s'$, which is the noisy sample at diffusion step $t_i$, as input and outputs the parameterization of a Gaussian distribution to get the reversed transition kernel $q(s, t_{i-1} | s, t_i)$. Each residue block consists of three components: two 1-d convolution operations with kernel size 3 and a group normalization between them.

The diffusion step $t_i$ is added to each convolutional block after being transformed by the sinusoidal position embedding. The final conv label denotes the convolution operation with kernel size equal to 1. Max-pooling reduces size of the features by half. Up-sampling uses the transposed convolution to expand the size of features.

The main idea behind the use of DDPM here is to learn a simple and easy-to-sample-from distribution $P_{\text{simple}}(s)$ that approximates the true $p(x, \beta)$ where $s = \{x, \beta\}$. DDPM does this by learning to reverse a gradual, multi-step noising process that starts with samples generated from the distribution $p(s)$, which we do not have direct access to, and diffuses to the simpler distribution $P_{\text{simple}}(s)$ that is easy-to-sample. For instance, $P_{\text{simple}}(s)$ could be an isotropic Gaussian. In addition to learning this noising process, DDPMs also learn the reverse denoising process which allows us to go back from samples generated using $P_{\text{simple}}(s)$ to samples that would have been generated from the underlying $P(s)$. Both the noising and denoising processes are modeled using diffusion processes that convert probability distributions to one another, and are implemented using the architecture shown in Fig. 1 which is based on the standard architecture for DDPMs as described in Ref. [18].

The noising diffusion process carried out in the space $s = \{x, \beta\}$ that converts $p(x, \beta)$ to the simpler $p_{simple}(x, \beta)$ can be decomposed into $M$ discrete steps denoted by corresponding transition probabilities $p(s', t_{i+1} | s, t_i)$ where $i \in [0, M]$, $P(s, t_0) \equiv p(x, \beta)$, and $P(s, t_M) \equiv p_{simple}(x, \beta)$. In DDPM, this noising diffusion process that converts sampled data to essentially noise, is set to be an Ornstein–Uhlenbeck (OU) process in which the tran-
position probability follows an simple Gaussian form. One can then easily generate samples from $P(s, t_0) \equiv p(x, \beta)$. The tricky bit now is to convert these samples back to the original distribution. In Ref. [17] it was shown that this transition reversed diffusion kernel $P(s, t_i|s', t_{i+1})$ can also be written in a Gaussian form. A deep neural network (Fig. 1) is then trained using variational inference to learn the approximate reversed transition kernel $Q(s, t_i|s', t_{i+1}) \approx P(s, t_i|s', t_{i+1})$. Thus by generating samples from a normal Gaussian distribution, which we can easily do in large numbers, and then passing these through the reversed transition kernel, we can generate samples that follow the target distribution $p(x, \beta)$ as desired. Note that instead of learning the joint probabilities $P(s_1, s_2)$, it can be advantageous to learn the conditional probability $P(s_1|s_2)$. This can be done through the protocol in Ref. [17] by adding a delta function to allow only a subset of $s$ to change during the noising and denoising process, i.e. $\delta(s_2 - s'_2)P(s_1, s_2, t_i|s'_1, s'_2, t_{i+1})$. This is very useful when for instance we are interested in generating samples only at a certain temperature or only in certain regions of the configuration space. In the most general form of diffusion probabilistic models (DPM) the reversed transition kernel $Q(s, t_i|s', t_{i+1})$ is considered as $Q(s, t - 1 | s', t) = \mathcal{N}(s; \tilde{\mu}(s', t), \tilde{\sigma}(s', t))$ and the neural network is trained to learn the mean $\tilde{\mu}(s', t)$ and variance $\tilde{\sigma}(s', t)$. In practice, however, there are many different ways to choose the Gaussian distribution parameterization. In “Denoising diffusion probabilistic models (DDPM)” [13] a new parameterization was designed to reduce the complexity of the training task. DDPM got its name because such a design makes the learning task resemble a denoising score matching procedure. [19] In Ref. [18] it was shown that with such a design, DDPM can generate samples of a quality that are comparable or even better than other generative models.

We now demonstrate how the above protocol can be applied to practical REMD. Here we study Aib$_9$ in explicit TIP3P water using CHARMM36M force-field. Aib$_9$ is a small peptide chain with 9 residues (Fig. 2a) that displays relatively rich complex conformational dynamics including the transitions between fully left-handed and right-handed helices. However, even in 4 $\mu$s unbiased MD at 400K, one can see only around 2–3 transitions between these two dominant equiprobable conformations, and even fewer, if any, transitions to the higher energy metastable states. To improve the sampling of this system, we perform REMD with 10 replicas at geometrically spaced temperatures ranging from 400K to 518K, with temperature increased by 3% for each replica. The attempt of exchanging configurations was made every 20 ps, with acceptance rate around

---

Figure 2: (a) AIB$_9$ in left-handed and right-handed structures in explicit TIP3P water. (b) Free energy profile of residues (take residue 5 as an example) with two ground state labeled as L, R and two excited state labeled as La, Ra. When all residues are in L states, the peptide chain is in left-handed state and similar for right-handed state.
Figure 3: Samples (dots) from REMD (left) and DDPM (right) at 400K. The scatters are colored by the free energy (contour lines, separated every 0.74 $k_bT$. (a) Projection of samples and free energy profile on dihedral angles of residue 5. (b) Projection of samples and free energy profile on dihedral angles of residue 8. DDPM was able to generate samples in states that are not present in the training dataset for both residues, indicated with thick black arrows in the right panels.

1% $\sim$ 2% between neighboring replicas, which is intentionally lower than what one usually has in REMD. This is because we want to show that even in the extreme cases where the number of atoms $N$ is so large that replicas do not have enough overlap, or if one wants to reduce the number of replicas to save computational resources, our DDPM can still do a decent job of complementing the REMD and reconstructing the true probability distribution at the low temperature of interest. To quantify the quality of sampling, we focus on the 18 dihedral angles corresponding to all 9 residues: $(\Phi_1, \Psi_1, \Phi_2, \Psi_2, \cdots \Phi_9, \Psi_9)$.

In order to evaluate the the performance of DDPM in improving estimates from REMD, we consider the following three challenging tasks:

1. Can we improve the sampling quality for the lowest-temperature replica with more accurate probability estimates?

2. Can we generate samples in low probability regions such as transition and metastable states and reliably estimate their free energies?

3. Can we generate samples at temperatures that are not included in the replica ladder?

As benchmarks, we ran unbiased MD at 400K with for 4$\mu$s and at 500K for 0.6$\mu$s. As shown in the Ramachandran plot in Fig. 2b, the free energy surface is mainly characterized by four metastable states: two ground states (L, R) and two excited states (La, Ra). The Ramachandran plots of all nine residues in the system look qualitatively similar but the middle residues of the peptide tend to be less flexible with higher energy barriers. We thus focus on the sampling for residues 5 and 8 as shown in Fig. 3.

We train our DDPM on a REMD trajectory with not yet sufficient sufficiently sampling at 400K and use it to generate a new set of samples at 400K. As shown in Fig. 3 the DDPM did take the information from other replicas and generate samples in states that are not present in the training dataset for both residues, indicated with thick black arrows in the right panels of Fig. 3. Specifically, for residue 5 we can see that the transition states between state R and state La, which were not being sampled in the 400K replica, are populated in samples from DDPM. For residue 8 the improvement is even more striking as the state Ra which was simply not sampled in REMD now gets nicely populated after DDPM. To further quantify the improvement gained due to DDPM, we compare the free energy differences between different configurations from both REMD and DDPM and compare that with the much longer reference unbiased MD at 400K. Specifically, we consider two cases: 1. $\Delta G_{L-Ra}$, which is the free energy difference between ground state L ($-2.2 < \Phi < -0.1, -1.6 < \Psi < 0.9$) and excited state Ra($-1.6 < \Phi < -0.2, 1.4 < \Psi < 3.14$); and 2. $\Delta G_{L-TS}$ the free energy difference between the ground state L ($-2.2 < \Phi < -0.1, -1.6 < \Psi < 0.9$) and a transition state ($-1.8 < \Phi < 0.6, 2.0 < \Psi < 0.9$).
Figure 4: Comparing $\Delta G$ calculated from samples of REMD, DDPM and long unbiased MD. See main text for precise definitions of various states. (a) shows the free energy difference between the ground state L (orange box) and excited state Ra (green box). Green empty crosses are reference values from 4 $\mu$s long unbiased MD at 400 K. Green filled plus signs show values after REMD and DDPM, while orange circles show values after REMD without DDPM. (b) shows the free energy difference between the ground state L (orange box) and the transition state (purple box) at 2 different temperatures, 400K and 500K.

As shown in Fig. 4, the free energy difference calculated from samples generated by DDPM are much closer to the that from the reference MD in general.

We now move to the third task from the list above, and test DDPM’s ability to get generate samples by interpolating or even extrapolating across temperatures not considered in the ladder of replicas. In the first example, we use it to generate samples at 500K, as in the training set, there is no replica with temperature 500K. As shown in Fig. 4b, the $\Delta G$ calculated from samples of DDPM is in good agreement with that from the reference MD at 500K. In the second example, we removed the samples from 400K replica in the training set and use it train a new model. The new model was used to generate samples at 400K. Even though in the training set, the lowest temperature of the heat bath is 412K, the model can make good predic-

tion of free energy difference between states as shown in Fig. 5.

In this letter, we have presented an AI-based approach to improve the sampling of REMD through a post-processing framework. The central idea is to not treat the temperature $\beta$ of the system as a parameter set by the heat bath but as a variable that fluctuates as per the kinetic energy of the system being studied. Our numerical examples validate that by learning $p(x, \beta)$ from REMD trajectories with DDPM can generally improve the quality of sampling at low temperatures and even generate samples in states where have not even been visited in the replicas and at temperatures not considered in the ladder of replicas.

AI-based methods are being increasingly used in the study of biomolecular systems with many complex, interdependent degrees of freedom. We specifically want to highlight here the devel-
development of flow-type methods, which have been shown to have the unprecedented ability to generate samples from very high dimensional space. Compared with other flow type models such as normalizing flow that use deterministic functions to map from an easy-to-sample distribution to target distribution, the stochastic nature of DDPM avoids the restriction of preserving the topology of configuration space and thus allows the learning of more complicated distributions. Furthermore, in DDPM, the design of the transition kernels reduce the learning task to just learning the mean of Gaussian kernels. This makes the training easier compared with other methods like stochastic normalizing flow which require learning more complicated transition kernels. It is worth mentioning here that a recent application of normalizing flows also attempts to enhance REMD sampling through somewhat similar ideas as ours. However in that work the machinery is used to directly affect the acceptance protocol in REMD while ours is a purely post-processing scheme.

We would also like to note that even though we only studied examples where we generated samples conditional on one variable $\beta$, DDPM should also be able to deal with the REST version of replica exchange where the solvent and solute are separately coupled to two heat baths with different temperatures. In this case we would use to generate samples from $P(s, \beta_1, \beta_2)$. More generally, if we consider DDPM as a powerful tool to learn complicated joint or equivalently conditional distributions from high-dimensional data, it also has the potential to solve problems that can be rephrased as maximizing likelihoods functions as have been done in Ref. 13,27,29. Specifically our work bears parallel with the T-WHAM approach of Ref. 13 which also works with a similar joint probability in configuration and energy space after binning them. While detailed comparisons using optimized codes will be subject of future work, here we highlight one important aspect of our work that might give it an edge over other methods mixing sampling from different temperatures – for this purpose we do not compute weights involving exponentials of the full system’s kinetic or potential energy. We finally point out that this work shows the possibility of learning generative models in the space of generic thermodynamic ensembles, by following the simple recipe that control parameters can also be viewed as fluctuating variables. As long as one is not in the thermodynamic limit – something we do not have to worry about in molecular simulations – this should be thus a practical and useful procedure for problems even beyond replica exchange molecular dynamics.

Code and data availability

Open-source code implementing this post-processing method that can be used to improve upon any already-performed REMD run will be released through https://github.com/
Acknowledgement
The authors would like to thank Alexander A. Alemi for initially giving us the idea of considering diffusion probabilistic models. We also thank Zachary Smith and Ziyue Zou for proofreading the manuscript. This work was supported by the National Science Foundation, Grant No. CHE-2044165 and used XSEDE Bridges through allocation TG-CHE180053, which is supported by National Science Foundation grant number ACI-1548562. Y.W. would like to thank NCI-UMD Partnership for Integrative Cancer Research for financial support.

References

(1) Sugita, Y.; Okamoto, Y. Chemical Physics Letters 1999, 314, 141 – 151.

(2) Hansmann, U. H. Chemical Physics Letters 1997, 281, 140–150.

(3) Abrams, C.; Bussi, G. Entropy 2014, 16, 163–199.

(4) Abel, R.; Wang, L.; Harder, E. D.; Berne, B.; Friesner, R. A. Accounts of chemical research 2017, 50, 1625–1632.

(5) Liu, P.; Kim, B.; Friesner, R. A.; Berne, B. Proceedings of the National Academy of Sciences 2005, 102, 13749–13754.

(6) Wang, L.; Friesner, R. A.; Berne, B. The Journal of Physical Chemistry B 2011, 115, 9431–9438.

(7) Ballard, A. J.; Jarzynski, C. Proceedings of the National Academy of Sciences 2009, 106, 12224–12229.

(8) Trebst, S.; Troyer, M.; Hansmann, U. H. The Journal of chemical physics 2006, 124, 174903.

(9) Nadler, W.; Hansmann, U. H. Physical Review E 2007, 76, 057102.

(10) Kim, J.; Keyes, T.; Straub, J. E. The Journal of chemical physics 2010, 132, 224107.

(11) Chodera, J. D.; Shirts, M. R. The Journal of chemical physics 2011, 135, 194110.

(12) Gil-Ley, A.; Bussi, G. Journal of chemical theory and computation 2015, 11, 1077–1085.

(13) Gallicchio, E.; Andrec, M.; Felts, A. K.; Levy, R. M. The Journal of Physical Chemistry B 2005, 109, 6722–6731.

(14) Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. International Conference on Medical image computing and computer-assisted intervention. 2015; pp 234–241.

(15) Wu, Y.; He, K. Group Normalization. Proceedings of the European Conference on Computer Vision (ECCV). 2018.

(16) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. Attention is all you need. Advances in neural information processing systems. pp 5998–6008.

(17) Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. International Conference on Machine Learning. 2015; pp 2256–2265.

(18) Ho, J.; Jain, A.; Abbeel, P. arXiv preprint arXiv:2006.11239 2020.

(19) Vincent, P. Neural computation 2011, 23, 1661–1674.

(20) Jorgensen, W. L.; Chandrasekhar, J.; Madura, J. D.; Impey, R. W.; Klein, M. L. The Journal of chemical physics 1983, 79, 926–935.

(21) Huang, J.; Rauscher, S.; Nawrocki, G.; Ran, T.; Feig, M.; De Groot, B. L.;
(22) Botan, V.; Backus, E. H.; Pfister, R.; Moretto, A.; Crisma, M.; Toniolo, C.; Nguyen, P. H.; Stock, G.; Hamm, P. Proceedings of the National Academy of Sciences 2007, 104, 12749–12754.

(23) Rezende, D.; Mohamed, S. Variational inference with normalizing flows. International conference on machine learning. 2015; pp 1530–1538.

(24) Noé, F.; Olsson, S.; Köhler, J.; Wu, H. Science 2019, 365.

(25) Wu, H.; Köhler, J.; Noé, F. arXiv preprint arXiv:2002.06707 2020.

(26) Dibak, M.; Klein, L.; Noé, F. arXiv preprint arXiv:2012.00429 2020.

(27) Shirts, M. R.; Chodera, J. D. The Journal of chemical physics 2008, 129, 124105.

(28) Wu, H.; Paul, F.; Wehmeyer, C.; Noé, F. Proceedings of the National Academy of Sciences 2016, 113, E3221–E3230.

(29) Rosta, E.; Hummer, G. Journal of chemical theory and computation 2015, 11, 276–285.