THE JOURNAL OF PHYSICAL CHEMISTRY A
A JOURNAL OF THE AMERICAN CHEMICAL SOCIETY

pubs.acs.org/JPCA

Article

Multiscale Reweighted Stochastic Embedding: Deep Learning of Collective Variables for Enhanced Sampling

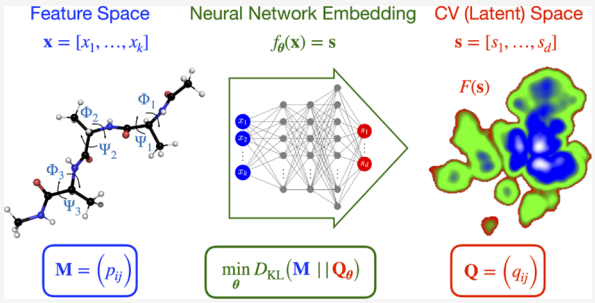
Jakub Rydzewski* and Omar Valsson*

Cite This: J. Phys. Chem. A 2021, 125, 6286–6302

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: Machine learning methods provide a general framework for automatically finding and representing the essential characteristics of simulation data. This task is particularly crucial in enhanced sampling simulations. There we seek a few generalized degrees of freedom, referred to as collective variables (CVs), to represent and drive the sampling of the free energy landscape. In theory, these CVs should separate different metastable states and correspond to the slow degrees of freedom of the studied physical process. To this aim, we propose a new method that we call multiscale reweighted stochastic embedding (MRSE). Our work builds upon a parametric version of stochastic neighbor embedding. The technique automatically learns CVs that map a high-dimensional feature space to a low-dimensional latent space via a deep neural network. We introduce several new advancements to stochastic neighbor embedding methods that make MRSE especially suitable for enhanced sampling simulations: (1) weight-tempered random sampling as a landmark selection scheme to obtain training data sets that strike a balance between equilibrium representation and capturing important metastable states lying higher in free energy; (2) a multiscale representation of the high-dimensional feature space via a Gaussian mixture probability model; and (3) a reweighting procedure to account for training data from a biased probability distribution. We show that MRSE constructs low-dimensional CVs that can correctly characterize the different metastable states in three model systems: the Müller-Brown potential, alanine dipeptide, and alanine tetrapeptide.



1. INTRODUCTION

Modeling the long-timescale behavior of complex dynamical systems is a fundamental task in the physical sciences. In principle, molecular dynamics (MD) simulations allow us to probe the spatiotemporal details of molecular processes, but the so-called sampling problem severely limits their usefulness in practice. This sampling problem comes from the fact that a typical free energy landscape consists of many metastable states separated by free energy barriers much higher than the thermal energy k_BT. Therefore, on the timescale one can simulate, barrier crossings are rare events, and the system remains kinetically trapped in a single metastable state.

One way to alleviate the sampling problem is to employ enhanced sampling methods.[1,2] In particular, one class of such methods works by identifying a few critical slow degrees of freedom, commonly referred to as collective variables (CVs), and then enhancing their fluctuations by introducing an external bias potential.[2−4] The performance of CV-based enhanced sampling methods depends heavily on the quality of the CVs. Effective CVs should discriminate between the relevant metastable states and include most of the slow degrees of freedom.[5] Typically, the CVs are selected manually by using physical and chemical intuition. Within the enhanced sampling community, numerous generally applicable CVs[1,6,7] have been developed and implemented in open-source codes.[8−10] However, despite immense progress in devising CVs, it may be far from trivial to find a set of CVs that quantify all the essential characteristics of a molecular system.

Machine learning (ML) techniques, in particular dimensionality reduction or representation learning methods,[11,12] provide a possible solution to this problem by automatically finding or constructing the CVs directly from the simulation data.[13−16] Such dimensionality reduction methods typically work in a high-dimensional feature space (e.g., distances, dihedral angles, or more intricate functions[17−19]) instead of directly using the microscopic coordinates, as this is much more efficient. Dimensionality reduction may employ linear or nonlinear transformations, for example, diffusion map,[20−23] stochastic neighbor embedding (SNE),[24−26] sketch-map,[27,28] and UMAP.[29] In the recent years, there has been a growing interest in performing nonlinear dimensionality reduction with deep neural networks (NNs) to provide parametric embeddings.

Received: March 30, 2021
Revised: June 17, 2021
Published: July 2, 2021

https://doi.org/10.1021/acs.jpca.1c02869
J. Phys. Chem. A 2021, 125, 6286–6302

Inspired by the seminal work of Ma and Dinner,[30] several such techniques recently applied to finding CVs include variational autoencoders,[31−34] time-lagged autoencoders,[35] symplectic flows,[36] stochastic kinetic embedding,[37] and encoder map.[38]

This work proposes a novel technique called multiscale reweighted stochastic embedding (MRSE) that unifies dimensionality reduction via deep NNs and enhanced sampling methods. The method constructs a low-dimensional representation of CVs by learning a parametric embedding from a high-dimensional feature space to a low-dimensional latent space. Our work builds upon various SNE methods.[24−26,39] We introduce several new aspects to SNE that makes MRSE particularly suitable for enhanced sampling simulations:

1. Weight-tempered random sampling as a landmark selection scheme to obtain training data sets that strike a balance between equilibrium representation and capturing important metastable states lying higher in free energy.
2. Multiscale representation of the high-dimensional feature space via a Gaussian mixture probability model.
3. Reweighting procedure to account for the sampling of the training data from a biased probability distribution.

We note that the overall objective of our research is to employ MRSE within an enhanced sampling scheme and improve the learned CVs iteratively. However, we focus mainly on the learning procedure for training data from enhanced sampling simulations in this work. Therefore, to eliminate the influence of possible incomplete sampling, we employ idealistic sampling conditions that are generally not achievable in practice.[40] To gauge the performance of the learning procedure and the quality of the resulting embeddings, we apply MRSE to three model systems (the Müller-Brown potential, alanine dipeptide, and alanine tetrapeptide) and provide a thorough analysis of the results.

2. METHODS

2.1. CV-Based Enhanced Sampling. We start by giving a theoretical background on CV-based enhanced sampling methods. We consider a molecular system, described by microscopic coordinates $\mathbf{R}$ and a potential energy function $U(\mathbf{R})$, which we want to study using MD or Monte Carlo simulations. Without loss of generality, we limit our discussion to the canonical ensemble (NVT). At equilibrium, the microscopic coordinates follow the Boltzmann distribution, $P(\mathbf{R}) = e^{-\beta U(\mathbf{R})}/\int d\mathbf{R}\, e^{-\beta U(\mathbf{R})}$, where $\beta = (k_B T)^{-1}$ is the inverse of the thermal energy.

In CV-based enhanced sampling methods, we identify a small set of coarse-grained order parameters that correspond to the essential slow degrees of freedom, referred to as CVs. The CVs are defined as $\mathbf{s}(\mathbf{R}) = [s_1(\mathbf{R}), s_2(\mathbf{R}),..., s_d(\mathbf{R})]$, where d is the number of CVs (i.e., the dimension of the CV space), and the dependence on $\mathbf{R}$ can be either explicit or implicit. Having defined the CVs, we obtain their equilibrium marginal distribution by integrating out all other degrees of freedom

$$P(\mathbf{s}) = \int d\mathbf{R}\, \delta[\mathbf{s} - \mathbf{s}(\mathbf{R})] P(\mathbf{R}) \tag{1}$$

where $\delta[\cdot]$ is the Dirac delta function. The integral in eq 1 is equivalent to $\langle \delta[\mathbf{s} - \mathbf{s}(\mathbf{R})] \rangle$, where $\langle \cdot \rangle$ denotes an ensemble average. Up to an unimportant constant, the free energy surface (FES) is given by $F(\mathbf{s}) = -\beta^{-1} \log P(\mathbf{s})$. In systems plagued by sampling problems, the FES consists of many metastable states separated by free energy barriers much larger than the thermal energy $k_B T$. Therefore, on the timescales we can simulate, the system stays kinetically trapped and is unable to explore the full CV space. In other words, barrier crossings between metastable states are rare events.

CV-based enhanced sampling methods overcome the sampling problem by introducing an external bias potential $V(\mathbf{s}(\mathbf{R}))$ acting in CV space. This leads to sampling according to a biased distribution

$$P_V(\mathbf{R}) = \frac{e^{-\beta[U(\mathbf{R}) + V(\mathbf{s}(\mathbf{R}))]}}{\int d\mathbf{R}\, e^{-\beta[U(\mathbf{R}) + V(\mathbf{s}(\mathbf{R}))]}} \tag{2}$$

We can trace this idea of non-Boltzmann sampling back to the seminal work by Torrie and Valleau published in 1977.[41] Most CV-based methods adaptively construct the bias potential on-the-fly during the simulation to reduce free energy barriers or even completely flatten them. At convergence, the CVs follow a biased distribution

$$P_V(\mathbf{s}) = \int d\mathbf{R}\, \delta[\mathbf{s} - \mathbf{s}(\mathbf{R})] P_V(\mathbf{R}) = \frac{e^{-\beta[F(\mathbf{s}) + V(\mathbf{s})]}}{\int d\mathbf{s}\, e^{-\beta[F(\mathbf{s}) + V(\mathbf{s})]}} \tag{3}$$

that is easier to sample. CV-based methods differ in how they construct the bias potential and which kind of biased CV sampling they obtain at convergence. A non-exhaustive list of modern CV-based enhanced sampling techniques includes multiple windows umbrella sampling,[42] adaptive biasing force,[43−45] Gaussian-mixture umbrella sampling,[46] metadynamics,[2,47,48] variationally enhanced sampling,[49,50] on-the-fly probability-enhanced sampling,[51,52] and ATLAS.[53] In the following, we focus on well-tempered metadynamics (WT-MetaD).[2,48] However, we can use MRSE with almost any CV-based enhanced sampling approach.

In WT-MetaD, the time-dependent bias potential is constructed by periodically depositing repulsive Gaussian kernels at the current location in CV space. Based on the previously deposited bias, the Gaussian height is scaled such that it gradually decreases over time.[48] In the long-time limit, the Gaussian height goes to zero. As has been proven,[54] the bias potential at convergence is related to the free energy by

$$V(\mathbf{s}, t \to \infty) = -\left(1 - \frac{1}{\gamma}\right) F(\mathbf{s}) \tag{4}$$

and we obtain a so-called well-tempered distribution for the CVs

$$P_V(\mathbf{s}) = \frac{[P(\mathbf{s})]^{1/\gamma}}{\int d\mathbf{s}\, [P(\mathbf{s})]^{1/\gamma}} \tag{5}$$

where $\gamma > 1$ is a parameter called bias factor that determines how much we enhance CV fluctuations. The limit $\gamma \to 1$ corresponds to the unbiased ensemble, while the limit $\gamma \to \infty$ corresponds to conventional (non-well-tempered) metadynamics.[47] If we take the logarithm of both sides of eq 5, we can see that sampling the well-tempered distribution is equivalent to sampling an effective FES, $F_\gamma(\mathbf{s}) = F(\mathbf{s})/\gamma$, where the barriers of the original FES are reduced by a factor of γ. In general, one should select a bias factor γ such that effective free energy barriers are on the order of the thermal energy $k_B T$.

Due to the external bias potential, each microscopic configuration $\mathbf{R}$ carries an additional statistical weight $w(\mathbf{R})$ that needs to be taken into account when calculating equilibrium

https://doi.org/10.1021/acs.jpca.1c02869
J. Phys. Chem. A 2021, 125, 6286−6302

properties. For a static bias potential, the weight is time-independent and given by $w(\mathbf{R}) = e^{\beta V(\mathbf{s}(\mathbf{R}))}$. In WT-MetaD, however, we need to take into account the time dependence of the bias potential, and thus, the weight is modified in the following way

$$w(\mathbf{R}, t) = \exp[\beta \tilde{V}(\mathbf{s}(\mathbf{R}), t)] \tag{6}$$

where $\tilde{V}(\mathbf{s}(\mathbf{R}), t) = V(\mathbf{s}(\mathbf{R}), t) - c(t)$ is the relative bias potential modified by introducing $c(t)$, a time-dependent constant that can be calculated from the bias potential at time t as[2,55]

$$c(t) = \frac{1}{\beta}\log \frac{\int d\mathbf{s} \exp\left[\frac{\gamma}{\gamma-1}\beta V(\mathbf{s}, t)\right]}{\int d\mathbf{s} \exp\left[\frac{1}{\gamma-1}\beta V(\mathbf{s}, t)\right]} \tag{7}$$

There are also other ways to reweight WT-MetaD simulations.[56−59]

In MD simulations, we do not only need to know the values of the CVs but also their derivatives with respect to the microscopic coordinates, $\nabla_{\mathbf{R}}\mathbf{s}(\mathbf{R})$. The derivatives are needed to calculate the biasing force $-\nabla_{\mathbf{R}}V(\mathbf{s}(\mathbf{R})) = -\partial_{\mathbf{s}}V(\mathbf{s})\cdot\nabla_{\mathbf{R}}\mathbf{s}(\mathbf{R})$. In practice, however, the CVs might not depend directly on $\mathbf{R}$, but rather indirectly through a set of some other input variables (e.g., features). We can even define a CV that is a chain of multiple variables that depend sequentially on each other. In such cases, it is sufficient to know the derivatives of the CVs with respect to the input variables, as we can obtain the total derivatives via the chain rule. In codes implementing CVs and enhanced sampling methods,[8−10] like PLUMED,[9,60] the handling of the chain rule is done automatically. Thus, when implementing a new CV, we only need to calculate its values and derivatives with respect to the input variables.

Having provided the basics of CV-based enhanced sampling simulations, we now introduce our method for learning CVs.

2.2. Multiscale Reweighted Stochastic Embedding. The basis of our method is the t-distributed variant of stochastic neighbor embedding (t-SNE),[25] a dimensionality reduction algorithm for visualizing high-dimensional data, for instance, generated by unbiased MD simulations.[61−64] We introduce here a parametric and multiscale variant of SNE aimed at learning CVs from atomistic simulations. In particular, we focus on using the method within enhanced sampling simulations, where we need to consider biased simulation data. We refer to this method as MRSE.

We consider a high-dimensional feature space, $\mathbf{x} = [x_1,..., x_k]$, of dimension k. The features could be distances, dihedral angles, or some more complex functions,[17−19] which depend on the microscopic coordinates. We introduce a parametric embedding function $f_{\boldsymbol{\theta}}(\mathbf{x}) = \mathbf{s}(\mathbf{x})$ that depends on parameters, $\boldsymbol{\theta}$, to map from the high-dimensional feature space to the low-dimensional latent space (i.e., the CV space), $\mathbf{s} = [s_1,..., s_d]$, of dimension d. From a molecular simulation, we collect N observations (or simply samples) of the features, $[\mathbf{x}_1,...,\mathbf{x}_N]^T$, that we use as training data. Using these definitions, the problem of finding a low-dimensional set of CVs amounts to using the training data to find an optimal parametrization for the embedding function given a nonlinear ML model. We can then use the embedding as CVs and project any point in feature space to CV space.

In SNE methods, this problem is approached by taking the training data and modeling the pairwise probability distributions for distances in the feature and latent space. To establish the notation, we write the pairwise probability distributions as $\mathbf{M} = (p_{ij})$ and $\mathbf{Q} = (q_{ij})$, where $1 \le i, j \le N$, for the feature and the latent space, respectively. For the pairwise probability distribution $\mathbf{M}$ ($\mathbf{Q}$), the interpretation of a single element p_{ij} (q_{ij}) is that higher the value, higher is the probability of picking $\mathbf{x}_j$ ($\mathbf{s}_j$) as a neighbor of $\mathbf{x}_i$ ($\mathbf{s}_i$). The mapping from the feature space to the latent space is then varied by adjusting the parameters $\boldsymbol{\theta}$ to minimize a loss function that measures the statistical difference between the two pairwise probability distributions. In the following, we explicitly introduce the pairwise probability distributions and the loss function used in MRSE.

2.2.1. Feature Pairwise Probability Distribution. We model the feature pairwise probability distribution for a pair of samples $\mathbf{x}_i$ and $\mathbf{x}_j$ from the training data as a discrete Gaussian mixture. Each term in the mixture is a Gaussian kernel

$$K_{\varepsilon_i}(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\varepsilon_i\|\mathbf{x}_i - \mathbf{x}_j\|_2^2) \tag{8}$$

that is characterized by a scale parameter ε_i associated to feature sample $\mathbf{x}_i$. A scale parameter is defined as $\varepsilon_i = 1/(2\sigma_i^2)$, where σ_i is the standard deviation (i.e., bandwidth) of the Gaussian kernel. Because $\varepsilon_i \neq \varepsilon_j$, the kernels are not symmetric. To measure the distance between data points, we employ the Euclidean distance $\|\cdot\|_2$ as an appropriate metric for representing high-dimensional data on a low-dimensional manifold.[65] Then, a pair $\mathbf{x}_i$ and $\mathbf{x}_j$ of points close to each other, as measured by the Euclidean distance, has a high probability of being neighbors.

For training data obtained from an enhanced sampling simulation, we need to correct the feature pairwise probability distribution because each feature sample $\mathbf{x}$ has an associated statistical weight $w(\mathbf{x})$. To this aim, we introduce a reweighted Gaussian kernel as

$$\tilde{K}_{\varepsilon_i}(\mathbf{x}_i, \mathbf{x}_j) = r(\mathbf{x}_i, \mathbf{x}_j)K_{\varepsilon_i}(\mathbf{x}_i, \mathbf{x}_j) \tag{9}$$

where $r(\mathbf{x}_i,\mathbf{x}_j) = \sqrt{w(\mathbf{x}_i)w(\mathbf{x}_j)}$ is a pairwise reweighting factor. As noted previously, the exact expression for the weights depends on the enhanced sampling method used. For training data from an unbiased simulation, or if we do not incorporate the weights into the training, all the weights are equal to one and $r(\mathbf{x}_i, \mathbf{x}_j) \equiv 1$ for $1 \le i, j \le N$.

A reweighted pairwise probability distribution for the feature space is then written as

$$\mathbf{P} = (p_{ij}^{\boldsymbol{\varepsilon}})_{1\le i,j\le N} \text{ and } p_{ij}^{\boldsymbol{\varepsilon}} = \frac{\tilde{K}_{\varepsilon_i}(\mathbf{x}_i, \mathbf{x}_j)}{\sum_k \tilde{K}_{\varepsilon_i}(\mathbf{x}_i, \mathbf{x}_k)} \tag{10}$$

with $p_{ii}^{\boldsymbol{\varepsilon}} = 0$. This equation represents the reweighted pairwise probability of features $\mathbf{x}_i$ and $\mathbf{x}_j$ for a given set of scale parameters $\boldsymbol{\varepsilon} = [\varepsilon_1, \varepsilon_2,..., \varepsilon_N]$, where each scale parameter is assigned to a row of the matrix $\mathbf{P}$. The pairwise probabilities $p_{ij}^{\boldsymbol{\varepsilon}}$ are not symmetric due to the different values of the scale parameters ($\varepsilon_i \neq \varepsilon_j$), which is in contrast to t-SNE, where the symmetry of the feature pairwise probability distribution is enforced.[25]

As explained in Section 2.2.3 below, the multiscale feature pairwise probability distribution $\mathbf{M}$ is written as a mixture of such pairwise probability distributions, each with a different set of scale parameters. In the next section, we describe how to calculate the scale parameters for the probability distribution given by eq 10.

2.2.2. Entropy of the Reweighted Feature Probability Distribution. The scale parameters $\boldsymbol{\varepsilon}$ used for the reweighted Gaussian kernels in eq 10 are positive scaling factors that need to be optimized to obtain a proper density estimation of the underlying data. We have that $\varepsilon_i = 1/(2\sigma_i^2)$, where σ_i is the standard deviation (i.e., bandwidth) of the Gaussian kernel.

https://doi.org/10.1021/acs.jpca.1c02869
J. Phys. Chem. A 2021, 125, 6286−6302

Therefore, we want a smaller σ_i in dense regions and a larger σ_i in sparse regions. To achieve this task, we define the Shannon entropy of the ith Gaussian probability as

$$H(\mathbf{x}_i) = -\sum_j p_{ij}^{\varepsilon_i} \log p_{ij}^{\varepsilon_i} \tag{11}$$

where the term $p_{ij}^{\varepsilon_i}$ refers to matrix elements from the ith row of $\mathbf{P}$ as eq 11 is solved for each row independently. We can write $p_{ij}^{\varepsilon_i} = \frac{1}{\overline{p}_i}\tilde{K}_{\varepsilon_i}(\mathbf{x}_i, \mathbf{x}_j)$, where $\overline{p}_i = \sum_k \tilde{K}_{\varepsilon_i}(\mathbf{x}_i, \mathbf{x}_k)$ is a row-wise normalization constant.

Inserting $p_{ij}^{\varepsilon_i}$ from eq 10 leads to the following expression

$$H(\mathbf{x}_i) = \log \overline{p}_i + \frac{\varepsilon_i}{\overline{p}_i}\sum_j \tilde{K}_{\varepsilon_i}(\mathbf{x}_i, \mathbf{x}_j)\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \underbrace{-\frac{1}{\overline{p}_i}\sum_j \tilde{K}_{\varepsilon_i}(\mathbf{x}_i, \mathbf{x}_j)\log r(\mathbf{x}_i, \mathbf{x}_j)}_{H_V(\mathbf{x}_i)} \tag{12}$$

where $H_V(\mathbf{x}_i)$ is a correction term due to the reweighting factor $r(\mathbf{x}_i, \mathbf{x}_j)$ introduced in eq 9. The reweighting factor is included also in the other two terms through $\tilde{K}_{\varepsilon_i}(\mathbf{x}_i,\mathbf{x}_j)$. For weights of the exponential form, like in WT-MetaD (eq 6), we have $w(\mathbf{x}_i) = e^{\beta V(\mathbf{x}_i)}$, and the correction term $H_V(\mathbf{x}_i)$ further reduces to

$$H_V(\mathbf{x}_i) = -\frac{\beta}{2}\left(\frac{1}{\overline{p}_i}\sum_j \tilde{K}_{\varepsilon_i}(\mathbf{x}_i, \mathbf{x}_j)V(\mathbf{x}_i) + V(\mathbf{x}_j)\right) \tag{13}$$

For the derivation of eqs 12 and 13, see Section S1 in Supporting Information.

For an unbiased simulation, or if we do not incorporate the weights into the training, is $r(\mathbf{x}_i, \mathbf{x}_j) \equiv 1$ for $1 \le i, j \le N$ and the correction term $H_V(\mathbf{x}_i)$ vanishes. Equation 12 then becomes $H(\mathbf{x}_i) = \log \overline{p}_i + \frac{\varepsilon_i}{\overline{p}_i}\sum_j K_{\varepsilon_i}(\mathbf{x}_i, \mathbf{x}_j)\|\mathbf{x}_i - \mathbf{x}_j\|_2^2$.

We use eq 12 to define an objective function for an optimization procedure that fits the Gaussian kernel to the data by adjusting the scale parameter so that $H(\mathbf{x}_i)$ is approximately $\log_2 PP$ (i.e., $\min_{\varepsilon_i}[H(\mathbf{x}_i) - \log_2 PP]$). Here PP is a model parameter that represents the perplexity of a discrete probability distribution. Perplexity is defined as an exponential of the Shannon entropy, $PP = 2^H$, and measures the quality of predictions for a probability distribution.[66] We can view the perplexity as the effective number of neighbors in a manifold.[25,26] To find the optimal values of the scale parameters, we perform the optimization using a binary search separately for each row of $\mathbf{P}$ (eq 10).

2.2.3. Multiscale Representation. As suggested in the work of Hinton and Roweis,[24] the feature probability distribution can be extended to a mixture, as done in refs 67−69. To this aim, for a given value of the perplexity PP, we find the optimal set of scale parameters $\boldsymbol{\varepsilon}^{PP}$ using eq 12. We do this for multiple values of the perplexity, $PP_l = 2^{L_{PP}-l+1}$, where l goes from 0 to $L_{PP} = \lfloor \log N \rfloor - 2$, and N is the size of the training data set. We then write the probabilities p_{ij} as an average over the different reweighted feature pairwise probability distributions

$$\mathbf{M} = (p_{ij})_{1\le i,j\le N} \text{ and } p_{ij} = \frac{1}{N_{PP}}\sum_{l=0}^{L_{PP}} p_{ij}^{\boldsymbol{\varepsilon}^{PP_l}} \tag{14}$$

where N_{PP} is the number of perplexities. Therefore, by taking p_{ij} as a Gaussian mixture over different perplexities, we obtain a multiscale representation of the feature probability distribution $\mathbf{M}$, without the need of setting perplexity by the user.

2.2.4. Latent Pairwise Probability Distribution. A known issue in many dimensionality reduction methods, including SNE, is the so-called "crowding problem",[24,70] which is caused partly by the curse of dimensionality.[71] In the context of enhanced sampling, the crowding problem would lead to the definition of CVs that inadequately discriminate between metastable states due to highly localized kernel functions in the latent space. As shown in Figure 1, if we change from a Gaussian kernel to a more heavy-tailed kernel for the latent space probability distribution, like a t-distribution kernel, we enforce that close-by data points are grouped while far-away data points are separated.

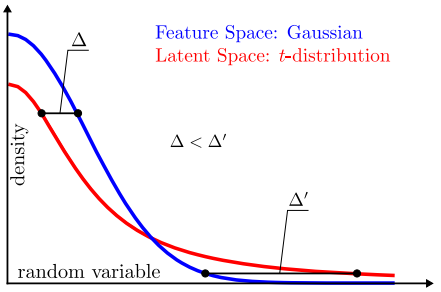


Figure 1. Schematic representation depicting how MRSE (and t-SNE) preserves the local structure of high-dimensional data. The pairwise probability distributions are represented by Gaussian kernels in the high-dimensional feature space and by the t-distribution kernels in the low-dimensional latent space. The minimization of the Kullback−Leibler (KL) divergence between the pairwise probability distributions enforces similar feature samples close to each other and separates dissimilar feature samples in the latent space. As the difference between the distributions fulfills $\Delta' > \Delta$, MRSE is likely to group close-by points into metastable states that are well separated.

Therefore, for the pairwise probability distribution in the latent space, we use a one-dimensional heavy-tailed t-distribution, which is the same as in t-SNE. We set

$$\mathbf{Q} = (q_{ij})_{1\le i,j\le N} \text{ and } q_{ij} = \frac{(1 + \|\mathbf{s}_i - \mathbf{s}_j\|_2^2)^{-1}}{\sum_k (1 + \|\mathbf{s}_i - \mathbf{s}_k\|_2^2)^{-1}} \tag{15}$$

where $q_{ii} = 0$ and the latent variables (i.e., the CVs) are obtained via the embedding function; for example, $\mathbf{s}_i = f_{\boldsymbol{\theta}}(\mathbf{x}_i)$.

2.2.5. Minimization of Loss Function. For the loss function to be minimized during the training procedure, we use the KL divergence $D_{\mathrm{KL}}(\mathbf{M}\|\mathbf{Q})$ to measure the statistical distance between the pairwise probability distributions M and $\mathbf{Q}$.[72] The loss function L for a data batch is defined as

$$D_{\mathrm{KL}}(\mathbf{M}\|\mathbf{Q}) = \frac{1}{N_b}\sum_{i=1}^{N_b}\sum_{\substack{i=1\\ i\neq j}}^{N_b} p_{ij}\log\left(\frac{p_{ij}}{q_{ij}}\right) \tag{16}$$

where $D_{\mathrm{KL}}(\mathbf{M}\|\mathbf{Q}) \ge 0$ with equality only when $\mathbf{M} = \mathbf{Q}$, and we split the training data into B batches of size N_b. We show the derivation of the loss function for the full set of N training data points in Section S2 in Supporting Information.

For the parametric embedding function $f_{\boldsymbol{\theta}}(\mathbf{x})$, we employ a deep NN (see Figure 2). After minimizing the loss function, we can use the parametric NN embedding function to project any given point in feature space to the latent space without rerunning the training procedure. Therefore, we can use the embedding as

https://doi.org/10.1021/acs.jpca.1c02869
J. Phys. Chem. A 2021, 125, 6286−6302

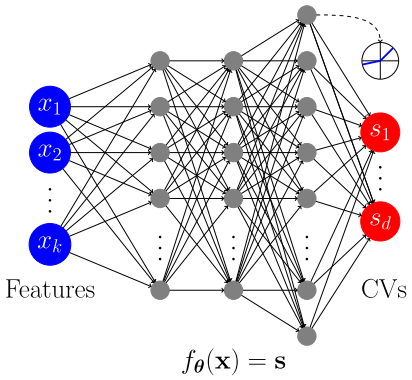


Figure 2. NN used to model the parametric embedding function $f_{\theta}(\mathbf{x})$. The input features $\mathbf{x}$, $\dim(\mathbf{x}) = k$ are fed into the NN to generate the output CVs $\mathbf{s}$, $\dim(\mathbf{s}) = d$. The parameters $\boldsymbol{\theta}$ represent the weights and biases of NN. The input layer is shown in blue, and the output layer is depicted in red. The hidden layers (gray) use dropout and leaky ReLU activations.

CVs, $\mathbf{s}(\mathbf{x}) = f_{\theta}(\mathbf{x})$. The derivatives of $f_{\theta}(\mathbf{x})$ with respect to $\mathbf{x}$ are obtained using backpropagation. Using the chain rule, we can then calculate the derivatives of $\mathbf{s}(\mathbf{x})$ with respect to the microscopic coordinates $\mathbf{R}$, which is needed to calculate the biasing force in an enhanced sampling simulation.

2.3. Weight-Tempered Random Sampling of Landmarks. A common way to reduce the size of a training set is to employ a landmark selection scheme before performing a dimensionality reduction.[73−76] The idea is to select a subset of the feature samples (i.e., landmarks) representing the underlying characteristics of the simulation data.

We can achieve this by selecting the landmarks randomly or with some given frequency in an unbiased simulation. If the unbiased simulation has sufficiently sampled phase space or if we use an enhanced sampling method that preserves the equilibrium distribution, like parallel tempering (PT),[77] the landmarks represent the equilibrium Boltzmann distribution. However, such a selection of landmarks might give an inadequate representation of transient metastable states lying higher in free energy, as they are rarely observed in unbiased simulations sampling the equilibrium distribution.

For simulation data resulting from an enhanced sampling simulation, we need to account for sampling from a biased distribution when selecting the landmarks. Thus, we take the statistical weights $w(\mathbf{R})$ into account within the landmark selection scheme. Ideally, we want the landmarks obtained from the biased simulation to strike a balance between an equilibrium representation and capturing higher-lying metastable states. Inspired by well-tempered farthest-point sampling (WT-FPS)[73] (see Section S3 in Supporting Information), we achieve this by proposing a simple landmark selection scheme appropriate for enhanced sampling simulations that we call weight-tempered random sampling.

In weight-tempered random sampling, we start by modifying the underlying data density by rescaling the statistical weights of the feature samples as $w(\mathbf{R}) \rightarrow [w(\mathbf{R})]^{1/\alpha}$. Here, $\alpha \geq 1$ is a tempering parameter similar in a spirit to the bias factor γ in the well-tempered distribution (eq 5). Next, we randomly sample landmarks according to the rescaled weights. This procedure results in landmarks distributed according to the following probability distribution

$$P_{\alpha}(\mathbf{x}) = \frac{\int \mathrm{d}\mathbf{R}\, [w(\mathbf{R})]^{1/\alpha} \delta[\mathbf{x} - \mathbf{x}(\mathbf{R})] P_V(\mathbf{R})}{\int \mathrm{d}\mathbf{R}\, [w(\mathbf{R})]^{1/\alpha} P_V(\mathbf{R})} \tag{17}$$

which we can rewrite as a biased ensemble average

$$P_{\alpha}(\mathbf{x}) = \frac{\langle [w(\mathbf{R})]^{1/\alpha} \delta[\mathbf{x} - \mathbf{x}(\mathbf{R})] \rangle_V}{\langle [w(\mathbf{R})]^{1/\alpha} \rangle_V} \tag{18}$$

Similar weight transformations have been used for treating weights degeneracy in importance sampling.[78]

For $\alpha = 1$, we recover weighted random sampling,[79] where we sample landmarks according to their unscaled weights $w(\mathbf{R})$. As we can see from eq 17, this should, in principle, give an equilibrium representation of landmarks, $P_{\alpha=1}(\mathbf{x}) = P(\mathbf{x})$. By employing $\alpha > 1$, we gradually start to ignore the underlying weights when sampling the landmarks and enhance the representation of metastable states lying higher in free energy. In the limit of $\alpha \rightarrow \infty$, we ignore the weights (i.e., all are equal to unity) and sample the landmarks randomly so that their distribution should be equal to the biased feature distribution sampled under the influence of the bias potential, $P_{\alpha\rightarrow\infty}(\mathbf{x}) = P_V(\mathbf{x})$. Therefore, the tempering parameter α allows us to tune the landmark selection between these two limits of equilibrium and biased representation. Using $\alpha > 1$ that is not too large, we can obtain a landmark selection that makes a trade-off between an equilibrium representation and capturing higher-lying metastable states.

To understand better the effect of the tempering parameter α, we can look at how the landmarks are distributed in the space of the biased CVs for the well-tempered case (eq 5). As shown in Section S4 in Supporting Information, we obtain

$$P_{\alpha}(\mathbf{s}) = \frac{[P(\mathbf{s})]^{1/\tilde{\alpha}}}{\int \mathrm{d}\mathbf{s}\, [P(\mathbf{s})]^{1/\tilde{\alpha}}} \tag{19}$$

where we introduce an effective tempering parameter $\tilde{\alpha}$ as

$$\tilde{\alpha} = \left(\frac{1}{\alpha} - \frac{1}{\alpha\gamma} + \frac{1}{\gamma}\right)^{-1} = \frac{\gamma\alpha}{\gamma + \alpha - 1} \tag{20}$$

that is unity for $\alpha = 1$ and goes to γ in the limit $\alpha \rightarrow \infty$. Thus, the effect of α is to broaden the CV distribution of the selected landmarks. In Figure 3, we show how the effective tempering parameter $\tilde{\alpha}$ depends on α for typical bias factor values γ.

The effect of α on the landmark feature distribution $P_{\alpha}(\mathbf{x})$ is harder to gauge as we cannot write the biased feature distribution $P_V(\mathbf{x})$ as a closed-form expression. In particular, for the well-tempered case, $P_V(\mathbf{x})$ is not given by $\propto [P(\mathbf{x})]^{1/\gamma}$, as

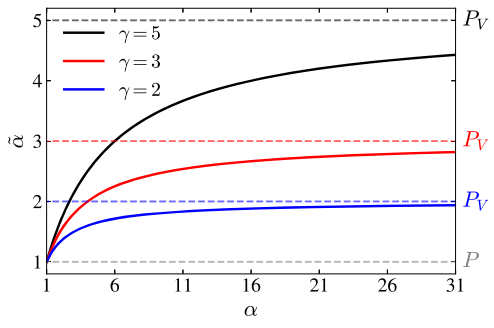


Figure 3. Effective tempering parameter $\tilde{\alpha}$ in the weight-tempered random sampling landmark selection scheme.

https://doi.org/10.1021/acs.jpca.1c02869
J. Phys. Chem. A 2021, 125, 6286−6302

the features are generally not fully correlated to the biased CVs.[80] The correlation of the features with biased CVs will vary greatly, also within the selected feature set. For features uncorrelated to the biased CVs, the biased distribution is nearly the same as the unbiased distribution. Consequently, the effect of tempering parameter α for a given feature will depend on the correlation with the biased CVs. In Section 4.2, we will show examples of this issue.

2.4. Implementation. We implement the MRSE method and the weight-tempered random sampling landmark selection method in an additional module called LowLearner in a development version (2.7.0-dev) of the open-source PLUMED[9,60] enhanced sampling plugin. The implementation is available openly at Zenodo[81] (DOI: 10.5281/zenodo.4756093) and from the PLUMED NEST[60] under plumID:21.023 at https://www.plumed-nest.org/eggs/21/023/. We use the LibTorch[82] library (PyTorch C++ API, git commit 89d6e88 used to obtain the results in this paper) that allows us to perform immediate execution of dynamic tensor computations with automatic differentiation.[83]

3. COMPUTATIONAL DETAILS

3.1. Model Systems. We consider three different model systems to evaluate the performance of the MRSE approach: the Müller-Brown Potential, alanine dipeptide, and alanine tetrapeptide. We use WT-MetaD simulations to generate biased simulation data sets used to train the MRSE embeddings for all systems. We also run unbiased simulation data sets for alanine di- and tetrapeptide by performing PT simulations that ensure proper sampling of the equilibrium distribution.

3.1.1. Müller-Brown Potential. We consider the dynamics of a single particle moving on the two-dimensional Müller-Brown potential,[84] $U(x, y) = \sum_j A_j e^{p_j(x,y)}$, where $p_j(x,y) = a_j(x - x_{0,j})^2 + b_j(x - x_{0,j})(y - y_{0,j}) + c_j(y - y_{0,j})^2$, x, y are the particle coordinates and $\mathbf{A}$, $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, $\mathbf{x}_0$ and $\mathbf{y}_0$ are the parameters of the potential given by $\mathbf{A} = (-40, -20, -34, 3)$, $\mathbf{a} = (-1, -1, 6.5, 0.7)$, $\mathbf{b} = (0, 0, 11, 0.6)$, $\mathbf{c} = (-10, -10, -6.5, -0.7)$, $\mathbf{x}_0 = (1, 0, -0.5, -1)$, and $\mathbf{y}_0 = (0, 0.5, 1.5, 1)$. Note that the $\mathbf{A}$ parameters are not the same as in ref 84 as we scale the potential to reduce the height of the barrier by a factor of 5. The FES as a function of the coordinates x and y is given directly by the potential, $F(x, y) = U(x, y)$. We employ rescaled units such that $k_B = 1$. We use the pesmd code from PLUMED[9,60] to simulate the system at a temperature of $T = 1$ using a Langevin thermostat[85] with a friction coefficient of 10 and employ a time step of 0.005. At this temperature, the potential has a barrier of around 20 k_BT between its two states and thus is a rare event system.

For the WT-MetaD simulations, we take x and y as CVs. We use different bias factors values (3, 4, 5, and 7), an initial Gaussian height of 1.2, a Gaussian width of 0.1 for both CVs, and deposit Gaussians every 200 steps. We calculate $c(t)$ (eq 7), needed for the weights, every time a Gaussian is added using a grid of 500^2 over the domain $[-5,5]^2$. We run the WT-MetaD simulations for a total time of 2×10^7 steps. We skip the first 20% of the runs (up to step 4×10^6) to ensure that we avoid the period at the beginning of the simulations where the weights might be unreliable due to rapid changes of the bias potential. For the remaining part, we normalize the weights such that they lie in the range 0 to 1 to avoid numerical issues.

We employ features saved every 1600 steps for the landmark selection data sets, yielding a total of 10^4 samples. From these data sets, we then use weight-tempered random sampling with $\alpha = 2$ to select 2000 landmarks that we use as training data to generate the MRSE embeddings.

For the embeddings, we use the coordinates x and y as input features ($k = 2$), while the number of output CVs is also 2 ($d = 2$). We do not standardize or preprocess the input features.

3.1.2. Alanine Dipeptide. We perform alanine dipeptide (Ace-Ala-Nme) simulations using the GROMACS 2019.2 code[86] patched with a development version of the PLUMED plugin.[9,60] We use the Amber99-SB force field[87] and a time step of 2 fs. We perform the simulations in the canonical ensemble using the stochastic velocity rescaling thermostat[88] with a relaxation time of 0.1 fs. We constrain hydrogen bonds using LINCS.[89] The simulations are performed in vacuum without periodic boundary conditions. We employ no cut-offs for electrostatic and non-bonded van der Waals interactions.

We employ four replicas with temperatures distributed geometrically in the range 300−800 K (300.0, 416.0, 576.9, and 800.0 K) for the PT simulation. We attempt exchanges between neighboring replicas every 10 ps. We run the PT simulation for 100 ns per replica. We only use the 300 K replica for analysis.

We perform the WT-MetaD simulations at 300 K using the backbone dihedral angles Φ and Ψ as CVs and employ different values for the bias factor (2, 3, 5, and 10). We use an initial Gaussian height of 1.2 kJ/mol, a Gaussian width of 0.2 rad for both CVs, and deposit Gaussians every 1 ps. We calculate $c(t)$ (eq 7) every time a Gaussian is added (i.e., every 1 ps) employing a grid of 500^2 over the domain $[-\pi,\pi]^2$. We run the WT-MetaD simulations for 100 ns. We skip the first 20 ns of the runs (i.e., first 20%) to ensure that we avoid the period at the beginning of the simulations where the weights might be unreliable due to rapid changes in the bias potential. For the remaining part, we normalize the weights such that they lie in the range 0−1 to avoid numerical issues.

For the landmark selection data sets, we employ features saved every 1 ps, which results in data sets of 8×10^4 and 1×10^5 samples for the WT-MetaD and PT simulations, respectively. We select 4000 landmarks for the training from these data sets, using weighted random sampling for the PT simulation and weight-tempered random sampling for the WT-MetaD simulations ($\alpha = 2$ unless otherwise specified).

For the embeddings, we use 21 heavy atoms pairwise distances as input features ($k = 21$) and the number of output CVs as 2 ($d = 2$). To obtain an impartial selection of features, we start with all 45 heavy-atom pairwise distances. Then, to avoid unimportant features, we automatically check for low variance features and remove all distances with a variance below 2×10^{-4} nm^2 from the training set (see Section S9 in Supporting Information). This procedure removes 24 distances and leaves 21 distances for the embeddings (both training and projections). We standardize remaining distances individually such that their mean is zero and their standard deviation is one.

3.1.3. Alanine Tetrapeptide. We perform simulations of alanine tetrapeptide (Ace-Ala_3-Nme) in vacuum using the GROMACS 2019.2 code[86] and a development version of the PLUMED plugin.[9,60] We use the same MD setup and parameters as for alanine dipeptide system, for example, the Amber99-SB force field;[87] see Section 3.1.2 for further details.

For the PT simulation, we employ eight replicas with temperatures ranging from 300 to 1000 K according to a geometric distribution (300.0, 356.4, 424.3, 502.6, 596.9, 708.9, 842.0, and 1000.0 K). We attempt exchanges between

https://doi.org/10.1021/acs.jpca.1c02869
J. Phys. Chem. A 2021, 125, 6286−6302

neighboring replicas every 10 ps. We simulate each replica for 100 ns. We only use the 300 K replica for analysis.

We perform the WT-MetaD simulation at 300 K using the backbone dihedral angles Φ_1, Φ_2, and Φ_3 as CVs and a bias factor of 5. We use an initial Gaussian height of 1.2 kJ/mol, a Gaussian width of 0.2 rad, and deposit Gaussians every 1 ps. We run the WT-MetaD simulation for 200 ns. We calculate $c(t)$ every 50 ps using a grid of 200^3 over the domain $[-\pi,\pi]^3$. We skip the first 40 ns of the run (i.e., first 20%) to ensure that we avoid the period at the beginning of the simulation where the weights are not equilibrated. We normalize the weights such that they lie in the range 0 to 1.

For the landmark selection data sets, we employ features saved every 2 ps for the WT-MetaD simulation and every 1 ps for the PT simulation. This results in data sets of 8×10^4 and 1×10^5 samples for the WT-MetaD and PT simulations, respectively. We select 4000 landmarks for the training from these data sets, using weighted random sampling for the PT simulation and weight-tempered random sampling with $\alpha = 2$ for the WT-MetaD simulations.

For the embeddings, we use sines and cosines of the dihedral angles (Φ_1, Ψ_1, Φ_2, Ψ_2, Φ_3, Ψ_3) as input features ($k = 12$), and the number of output CVs is 2 ($d = 2$). We do not standardize or preprocess the input features further.

3.2. NN Architecture. For the NN, we use the same size and number of layers as in the work of van der Maaten and Hinton.[26,90] The NN consists of an input layer with a size equal to the dimension of the feature space k, followed by three hidden layers of sizes $h_1 = 500$, $h_2 = 500$, and $h_3 = 2000$, and an output layer with a size equal to the dimension of the latent space d.

To allow for any output value, we do not wrap the output layer within an activation function. Moreover, for all hidden layers, we employ leaky rectified linear units (leaky ReLU)[91] with a leaky parameter set to 0.2. Each hidden layer is followed by a dropout layer[92] (dropout probability $p = 0.1$). For the details regarding the architecture of NNs, see Table 1.

3.3. Training Procedure. We shuffle the training data sets and divide them into batches of size 500. We initialize all trainable weights of the NNs with the Glorot normal scheme[93] using the gain value calculated for leaky ReLU. The bias parameters of the NNs are initialized with 0.005.

We minimize the loss function given by eq 16 using the Adam optimizer[94] with AMSGrad,[95] where we use learning rate $\eta = 10^{-3}$ and momenta $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We also employ a standard L2 regularization term on the trainable network parameters in the form of weight decay set to 10^{-4}. We perform the training for 100 epochs in all cases. The loss function learning curves for the systems considered here are shown in Section S7 in Supporting Information.

We report all hyperparameters used to obtain the results in this work in Table 1. For reproducibility purposes, we also list the random seeds used while launching the training runs (the seed affects both the landmark selection and the shuffling of the landmarks during the training).

3.4. Kernel Density Estimation. We calculate FESs for the trained MRSE embeddings using kernel density estimation (KDE) with Gaussian kernels. We employ a grid of 200^2 for the FES figures. We choose the bandwidths for each simulation data set by first estimating them using Silverman's rule and then adjusting the bandwidths by comparing the KDE FES to an FES obtained with a discrete histogram. We show a representative comparison between KDE and discrete FESs in Section S6 in Supporting Information. We employ reweighting for FESs from WT-MetaD simulation data where we weigh each Gaussian KDE kernel by the statistical weight $w(\mathbf{R})$ of the given data point.

3.5. Data Availability. The data supporting the results of this study are openly available at Zenodo[81] (DOI: 10.5281/zenodo.4756093). PLUMED input files and scripts required to replicate the results presented in the main text are available from the PLUMED NEST[60] under plumID:21.023 at https://www.plumed-nest.org/eggs/21/023/.

Table 1. Hyperparameters Used to Obtain the Results Reported in This Paper

hyperparameter	Müller-Brown	alanine dipeptide	alanine tetrapeptide
features	x and y	heavy atom distances	dihedral angles (cos/sin)
NN architecture	[2, 500, 500, 2000, 2]	[21, 500, 500, 2000, 2]	[12, 500, 500, 2000, 2]
optimizer	Adam (AMSGrad)	Adam (AMSGrad)	Adam (AMSGrad)
number of landmarks	$N = 2000$	$N = 4000$	$N = 4000$
batch size	$N_b = 500$	$N_b = 500$	$N_b = 500$
training iterations	100	100	100
learning rate	$\eta = 10^{-3}$	$\eta = 10^{-3}$	$\eta = 10^{-3}$
seed	111	111 (SI: 222, 333)	111
leaky parameter	0.2	0.2	0.2
dropout	$p = 0.1$	$p = 0.1$	$p = 0.1$
weight decay	10^{-4}	10^{-4}	10^{-4}
β_1, β_2	0.9 and 0.999	0.9 and 0.999	0.9 and 0.999

4. RESULTS

4.1. Müller-Brown Potential. We start by considering a single particle moving on the two-dimensional Müller-Brown potential, as shown in Figure 4a. We use this system as a simple test to check if the MRSE method can preserve the topography of the FES in the absence of any dimensionality reduction when performing a mapping with a relatively large NN.

We train the MRSE embeddings on simulation data sets obtained from WT-MetaD simulations using the coordinates x and y as CVs. Here, we show only the results obtained with bias factor $\gamma = 5$, while the results for other values are shown in Section S8 in Supporting Information. The MRSE embeddings can be freely rotated, and overall rotation is largely determined by the random seed used to generate the embeddings. Therefore, to facilitate comparison, we show here results obtained using the Procrustes algorithm to find an optimal rotation of the MRSE embeddings that best aligns with the original coordinates x and y. The original non-rotated embeddings are shown in Section S8 in Supporting Information. We present the FESs obtained with the MRSE embeddings in Figure 4b,c. We can see that the embeddings preserve the topography of the FESs very well and demonstrate a fine separation of metastable states, both when we incorporate the weights into the training through eq 9 (panel b), and when we do not (panel c).

To quantify the difference between the x and y coordinates and the CVs found by MRSE, we normalize all coordinates and plot CV_1 as a function of x and CV_2 as a function of y. In Figure 5, we can see that the points lie along the identity line, which shows

https://doi.org/10.1021/acs.jpca.1c02869
J. Phys. Chem. A 2021, 125, 6286−6302

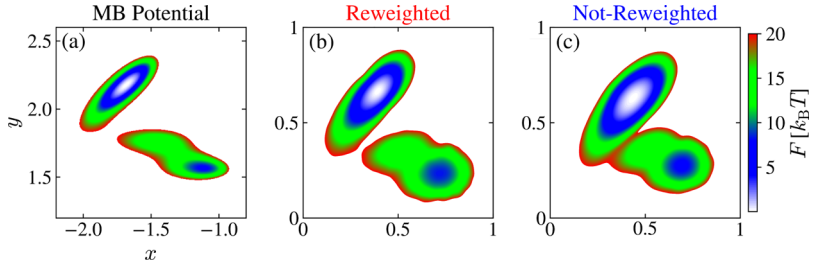


Figure 4. Results for the Müller-Brown potential. FESs for MRSE embeddings obtained from the WT-MetaD simulation (γ = 5). We show MRSE embeddings obtained with (b) and without (c) incorporating weights into the training via a reweighted feature pairwise probability distribution (see eq 9). The units for the MRSE embeddings are arbitrary and only shown as a visual guide. To facilitate comparison, we post-process the MRSE embeddings using the Procrustes algorithm to find an optimal rotation that best aligns with the original coordinates x and y; see text.

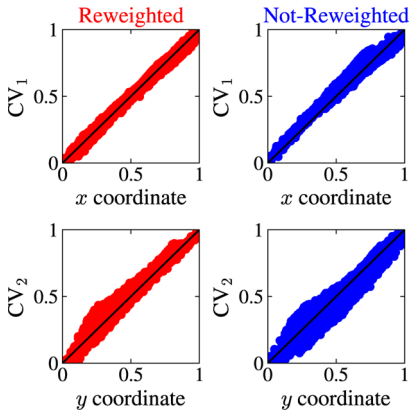


Figure 5. Results for the Müller-Brown potential. We show how the MRSE embeddings map the coordinates x and y by plotting the normalized coordinates x and y versus the normalized MRSE CVs. The MRSE embeddings are trained using data from a WT-MetaD simulation with γ = 5, and obtained with (red) and without (blue) incorporating weights into the training via a reweighted feature pairwise probability distribution (see eq 9). To facilitate comparison, we post-process the MRSE embeddings using the Procrustes algorithm to find an optimal rotation that best aligns with the original coordinates x and y; see text.

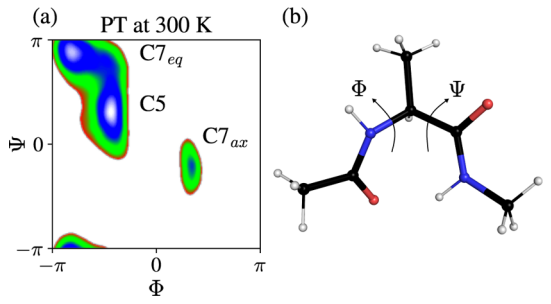


Figure 6. Results for alanine dipeptide in vacuum at 300 K. (a) Free energy landscape $F(\Phi, \Psi)$ from the PT simulation. The metastable states $C7_{eq}$, C5, and $C7_{ax}$ are shown. (b) Molecular structure of alanine dipeptide with the dihedral angles Φ and Ψ indicated.

that both MRSE embeddings preserve well the original coordinates of the MB system. In other words, the embeddings maintain the normalized distances between points. We analyze this aspect in a detailed manner for a high-dimensional set of features in Section 4.2.

4.2. Alanine Dipeptide. Next, we consider alanine dipeptide in vacuum, a small system often used to benchmark free energy and enhanced sampling methods. The free energy landscape of the system is described by the backbone (Φ, Ψ) dihedral angles. Generally, the (Φ, Ψ) angles are taken as CVs for biasing, as we do here to generate the training data set. However, for this particular setup in vacuum, it is sufficient to bias Φ to drive the sampling between states as Ψ is a fast CV compared to Φ. We can see in Figure 6 that three metastable states characterize the FES. The $C7_{eq}$ and C5 states are separated only by a small barrier of around 1−2 k_BT, so transitions between these two states are frequent. The $C7_{ax}$ state lies higher in free energy (i.e., is less probable to sample) and is separated by a high barrier of around 14 k_BT from the other two states; so transitions from $C7_{eq}$/C5 to $C7_{ax}$ are rare.

For the MRSE embeddings, we do not use the (Φ, Ψ) angles as input features, but rather a set of 21 heavy atom pairwise distances that we impartially select as described in Section 3.1.2. Using only the pairwise distances as input features makes the exercise of learning CVs more challenging as the Φ and Ψ angles cannot be represented as linear combinations of the interatomic distances. We can assess the quality of our results by examining how well the MRSE embeddings preserve the topography of the FES on local and global scales. However, before presenting the MRSE embeddings, let us consider the landmark selection, which we find crucial to our protocol to construct embeddings accurately.

As discussed in Section 2.3, we need to have a landmark selection scheme that takes into account the weights of the configurations and gives a balanced selection that ideally is close to the equilibrium distribution but represents all metastable states of the system, also the higher-lying ones. We devise for this task a method called weight-tempered random sampling. This method has a tempering parameter α that allows us to interpolate between an equilibrium and a biased representation of landmarks (see eq 17).

The effect of the tempering parameter α on the landmark feature distribution $P_\alpha(\mathbf{x})$ will depend on the correlation of the features with the biased CVs. The correlation will vary greatly, also within the selected feature set. In Figure 7, we show the marginal distributions for two examples from the feature set. For a feature correlated with the biased CVs, the biasing enhances the fluctuations, and we observe a significant difference between the equilibrium distribution and the biased one, as expected. In this case, the effect of introducing α is to interpolate between these two limits. On the other hand, for a feature not correlated to the biased CVs, the equilibrium and biased distribution are almost the same, and α does not affect the distribution of this feature.

In Figure 8, we show the results from the landmark selection for one of the WT-MetaD simulations (γ = 5). In the top row, we show how the selected landmarks are distributed in the CV

https://doi.org/10.1021/acs.jpca.1c02869
J. Phys. Chem. A 2021, 125, 6286−6302

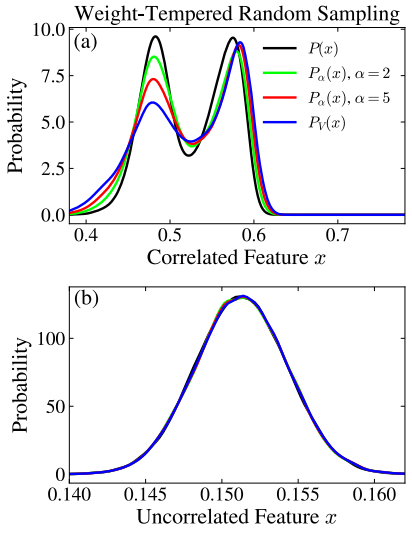


Figure 7. Results for alanine dipeptide in vacuum at 300 K. The effect of the tempering parameter α in the weight-tempered random sampling landmark selection scheme for a WT-MetaD simulation ($\gamma = 5$) biasing (Φ, Ψ). Marginal landmark distributions for two examples of features (i.e., heavy atom distances) from the feature set that are (a) correlated and (b) uncorrelated with the biased CVs. The units are nm.

space. In the bottom row, we show the effective FES of selected landmarks projected on the Ψ dihedral angle.

For $\alpha = 1$, equivalent to weighted random sampling,[76] we can see that we get a worse representation of the $C7_{ax}$ state as compared to the other states. We can understand this issue by considering the weights of configurations in the $C7_{ax}$ that are considerably smaller than the weights from the other states. As shown in Section S10 in Supporting Information, using the $\alpha = 1$ landmark results in an MRSE embedding close to the equilibrium PT embedding (shown in Figure 10a below) but has a worse separation of the metastable states as compared to other embeddings.

On the other hand, if we use $\alpha = 2$, we obtain a much more balanced landmark selection that is relatively close to the equilibrium distribution but has a sufficient representation of the $C7_{ax}$ state. Using larger values of α renders a selection closer to the sampling from the underlying biased simulation, with more features higher in free energy. We observe that using $\alpha = 2$ gives the best MRSE embedding. In contrast, higher values of α result in worse embeddings characterized by an inadequate mapping of the $C7_{ax}$ state, as can be seen in Section S12 in Supporting Information. Therefore, in the following, we use a value of $\alpha = 2$ for the tempering parameter in the landmark selection. This value corresponds to an effective landmark CV distribution broadening of $\tilde{\alpha} \approx 1.67$ (see eqs 19 and 20).

These landmark selection results underline the importance of having a balanced selection of landmarks that is close to the equilibrium distribution and gives a proper representation of all metastable states but excludes points from unimportant higher-lying free energy regions. The exact value of α that achieves such optimal selection will depend on the underlying free energy landscape.

In Section S11 in Supporting Information, we show results obtained using WT-FPS for the landmark selection (see Section S3 in Supporting Information for a description of WT-FPS). We can observe that the WT-MetaD embeddings obtained using WT-FPS with $\alpha = 2$ are similar to the WT-MetaD embeddings shown, as in Figure 10 below. Thus, for small values of the tempering parameter, both methods give similar results.

Having established how to perform the landmark selection, we now consider the results for MRSE embeddings obtained on unbiased and biased simulation data at 300 K. The unbiased simulation data comes from a PT simulation that accurately captures the equilibrium distribution within each replica.[77] Therefore, for the 300 K replica used for the analysis and training, we obtain the equilibrium populations of the different metastable states while not capturing the higher-lying and transition-state regions. In principle, we could also include simulation data from the higher-lying replica into the training by considering statistical weights to account for the temperature difference, but this would defeat the purpose of using the PT to generate unbiased simulation data that does not require reweighting. We refer to the embedding trained on the PT simulation data as the PT embedding. The biased simulation data comes from WT-MetaD simulations where we bias the (Φ, Ψ) angles. We refer to these embeddings as the WT-MetaD embeddings.

In the WT-MetaD simulations, we use bias factors from 2 to 10 to generate training data sets representing a biased

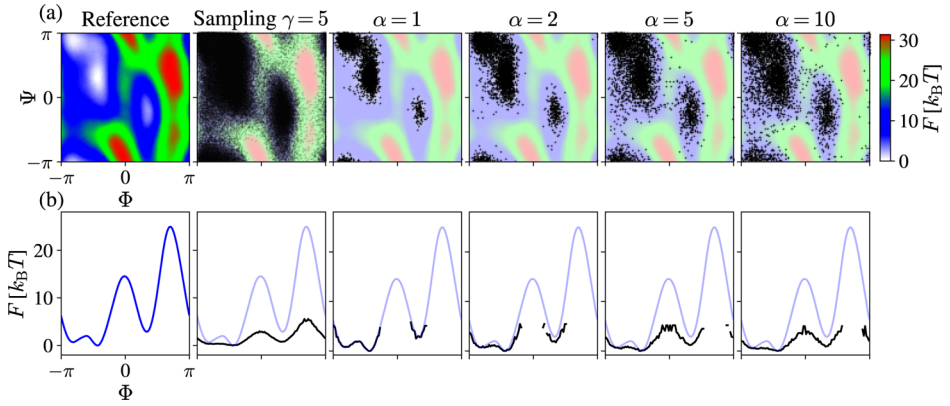


Figure 8. Results for alanine dipeptide in vacuum at 300 K. Weight-tempered random sampling as a landmark selection scheme for a WT-MetaD simulation ($\gamma = 5$) biasing (Φ, Ψ). (a) In the first two panels, we show the reference FES in the (Φ, Ψ) space and the points sampled during the simulations. In the subsequent panels, we present the 4000 landmarks selected for different values of the α parameter. (b) In the bottom row, we show the results projected on Φ, where the reference FES is shown in light blue. The projections (black) are calculated as a negative logarithm of the histogram of the selected landmarks.

https://doi.org/10.1021/acs.jpca.1c02869
J. Phys. Chem. A 2021, 125, 6286−6302

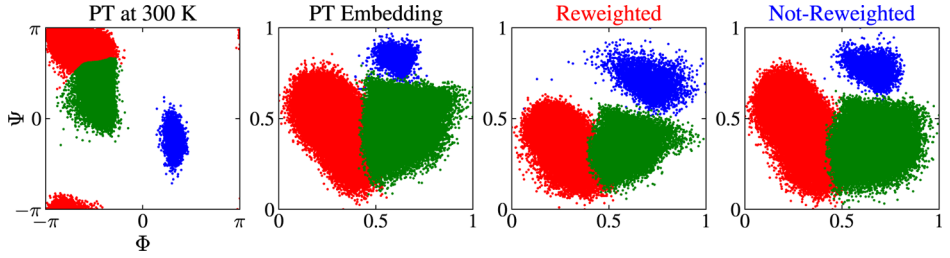


Figure 9. Results for alanine dipeptide in vacuum at 300 K. Clustering of the PT simulation data for the different embeddings. The results show how the embeddings map the metastable states. The data points are colored accordingly to their cluster. The first panel shows the metastable state clusters in the (Φ, Ψ) space. The second panel shows the results for the PT embedding. The third and fourth panels show the results for a representative case of a WT-MetaD embedding ($\gamma = 5$), obtained with and without incorporating weights into the training via a reweighted feature probability distribution (see eq 9), respectively. For the details about clustering,[96] see Section S5 in theSupporting Information. The units for the MRSE embeddings are arbitrary and only shown as a visual guide.

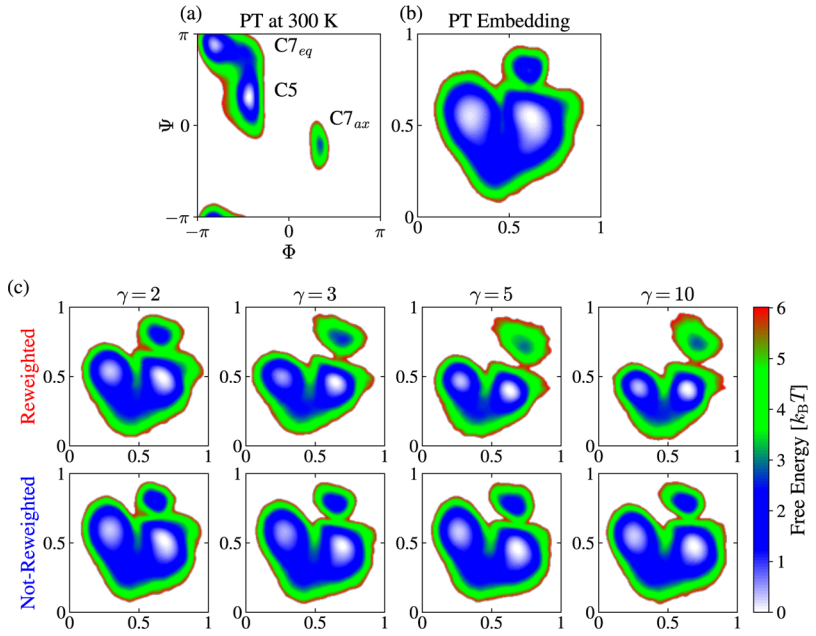


Figure 10. Results for alanine dipeptide in vacuum at 300 K. MRSE embeddings trained on unbiased and biased simulation data. (a) Free energy landscape $F(\Phi, \Psi)$ from the PT simulation. The metastable states $C7_{eq}$, C5, and $C7_{ax}$ are shown. (b) FES for the MRSE embedding trained using the PT simulation data. (c) FESs for the MRSE embeddings trained using the WT-MetaD simulation data. We show results obtained from the simulations using different bias factors γ. We show WT-MetaD embeddings obtained with (top row) and without (bottom row) incorporating weights into the training via a reweighted feature pairwise probability distribution (see eq 9). We obtain all the FESs by calculating the embeddings on the PT simulation data and using KDE as described in Section 3.4. The units for the MRSE embeddings are arbitrary and only shown as a visual guide.

distribution that progressively goes from a distribution closer to the equilibrium one to more flatter distribution as we increase γ (see eq 5). In this way, we can test how the MRSE training and reweighting procedure works when handling simulation data obtained under different biasing strengths.

For the WT-MetaD training data sets, we also investigate the effect of not incorporating the weight into the training via a reweighted feature pairwise probability distribution (i.e., all weights equal to unity in eq 9). In this case, only the weight-tempered random sampling landmark selection takes the weights into account. In the following, we refer to these WT-MetaD embeddings as without reweighting or not-reweighted.

To be consistent and allow for a fair comparison between embeddings, we evaluate all the trained WT-MetaD embeddings on the unbiased PT simulation data and use the resulting projections to perform analysis and generate FESs. This procedure is possible as both the unbiased PT and the biased WT-MetaD simulations sample all metastable states of alanine dipeptide (i.e., the WT-MetaD simulations do not sample metastable states that the PT simulation does not).

To establish that the MRSE embeddings correctly map the metastable states, we start by considering the clustering results in Figure 9. We can see that the PT embedding (second panel) preserves the topography of the FES and correctly maps all the important metastable states. We can say the same for the reweighted (third panel) and not-reweighted (fourth panel) embeddings. Thus, the embeddings map both the local and global characteristics of the FES accurately. Next, we consider the MRSE embeddings for the different bias factors.

In Figure 10, we show the FESs for the different embeddings along with the FES for the Φ and Ψ dihedral angles. For the reweighted WT-MetaD embeddings (top row of panel c), we can observe that all the embeddings are of consistent quality and exhibit a clear separation of the metastable states. In contrast, we can see that the not-reweighted WT-MetaD embeddings (bottom row of panel c) have a slightly worse separation of

https://doi.org/10.1021/acs.jpca.1c02869
J. Phys. Chem. A 2021, 125, 6286−6302

the metastable states. Thus, we can conclude that incorporating the weights into the training via a reweighted feature pairwise probability distribution (see eq 9) improves the visual quality of the embeddings for this system.

To further check the quality of the embeddings, we calculate the free energy difference between metastable states as $\Delta F_{A,B} = -\frac{1}{\beta}\log\left(\int_A d\mathbf{s}\, e^{-\beta F(\mathbf{s})} / \int_B d\mathbf{s}\, e^{-\beta F(\mathbf{s})}\right)$, where the integration domains are the regions in CV space corresponding to the states A and B, respectively. This equation is only valid if the CVs correctly discriminate between the different metastable states. For the MRSE embeddings, we can thus identify the integration regions for the different metastable states in the FES and calculate the free energy differences. Reference values can be obtained by integrating the $F(\Phi,\Psi)$ FES from the PT simulation. A deviation from a reference value would indicate that an embedding does not correctly map the density of the metastable states. In Figure 11, we show the free energy differences for all the MRSE embeddings. All free energy differences obtained with the MRSE embeddings agree with the reference values within a 0.1 k_BT difference for both reweighted and not-reweighted WT-MetaD embeddings. For bias factors larger than 3, we can observe that the reweighted embeddings perform distinctly better than the not-reweighted ones.

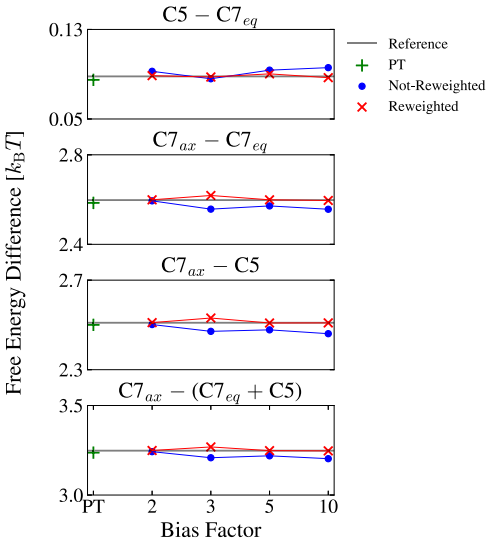


Figure 11. Results for alanine dipeptide in vacuum at 300 K. Free energy differences between metastable states for the FESs of the embeddings, as shown in Figure 10. We show the reference values from the $F(\Phi,\Psi)$ FES obtained from the PT simulation at 300 K as horizontal gray lines. The results for the reweighted embeddings are shown as red crosses, while the results for the not-reweighted embeddings are shown as blue dots. The results for the PT embedding are shown as green plus symbols.

As a final test of the MRSE embeddings for this system, we follow the approach used by Tribello and Gasparotto.[75,76] We calculate the pairwise distances between points in the high-dimensional feature space and the corresponding pairwise distances between points in the low-dimensional latent (i.e., CV) space given by the embeddings. We then calculate the joint probability density function of the distances using histogramming. The joint probability density should be concentrated on the identity line if an embedding preserves distances accurately. However, this only is valid for the MRSE embeddings constructed without incorporating the weights into the training, since for this case, there are no additional constraints besides geometry.

As we can see in Figure 12, the joint density is concentrated close to the identity line for most cases. For the reweighted WT-MetaD embeddings (panel b), the density for the distances in the middle range slightly deviates from the identity line in contrast to the not-reweighted embeddings. This deviation is due to additional constraints on the latent space. In the reweighted cases, apart from the Euclidean distances, we also include the statistical weights into the construction of the feature pairwise probability distribution. Consequently, having landmarks with low weights in the feature space decreases the probability of being neighbors to these landmarks in the latent space. Therefore, the deviation from the identity line must be higher for the reweighted embeddings.

Summarizing the results in this section, we can observe that MRSE can construct embeddings, both from unbiased and biased simulation data, that correctly describe the local and global characteristics of the free energy landscape of alanine dipeptide. For the biased WT-MetaD simulation data, we have investigated the effect of not including the weights in the training of the MRSE embeddings. Then, only the landmark selection takes the weights into account. The not-reweighted embeddings are similar or slightly worse than the reweighted ones. We can explain the slight difference between the reweighted and not-reweighted embeddings by that the weight-tempered random sampling does the primary reweighting. Nevertheless, we can conclude that incorporating the weights into the training is beneficial for alanine dipeptide test case.

4.3. Alanine Tetrapeptide. As the last example, we consider alanine tetrapeptide, a commonly used test system for enhanced sampling methods.[51,53,97−101] Alanine tetrapeptide is a considerably more challenging test case than alanine dipeptide. Its free energy landscape consists of many metastable states, most of which are high in free energy and thus difficult to capture in an unbiased simulation. We anticipate that we can only obtain an embedding that accurately separates all of the metastable states by using training data from an enhanced sampling simulation, which better captures higher-lying metastable states. Thus, the system is a good test case to evaluate the performance of the MRSE method and the reweighting procedure.

As it is often customary,[51,53,97,98] we consider the backbone dihedral angles $\mathbf{\Phi} \equiv (\Phi_1, \Phi_2, \Phi_3)$ and $\mathbf{\Psi} \equiv (\Psi_1, \Psi_2, \Psi_3)$ that characterize the configurational landscape of alanine tetrapeptide. We show the dihedral angles in Figure 13b. For this particular setup in vacuum, it is sufficient to use $\mathbf{\Phi}$ to describe the free energy landscape and separate the metastable states, as $\mathbf{\Psi}$ are fast CVs in comparison to $\mathbf{\Phi}$.[51,97] To generate biased simulation data, we perform WT-MetaD simulation using the $\mathbf{\Phi}$ angles as CVs and a bias factor $\gamma = 5$. Moreover, we perform a PT simulation and employ the 300 K replica to obtain unbiased simulation data. As before, the embeddings obtained by training on these simulation data sets are denoted as WT-MetaD and PT embeddings, respectively. As before, we also consider a WT-MetaD embedding, denoted as not-reweighted, where we do not include the weights into the construction of the feature pairwise probability distribution.

To verify the quality of the sampling and the accuracy of the FESs, we compare the results obtained from the WT-MetaD and PT simulations to results from bias-exchange metadynamics simulations[102] using $\mathbf{\Phi}$ and $\mathbf{\Psi}$ as CVs (see Section S13 in Supporting Information). Comparing the free energy profiles for

https://doi.org/10.1021/acs.jpca.1c02869
J. Phys. Chem. A 2021, 125, 6286−6302

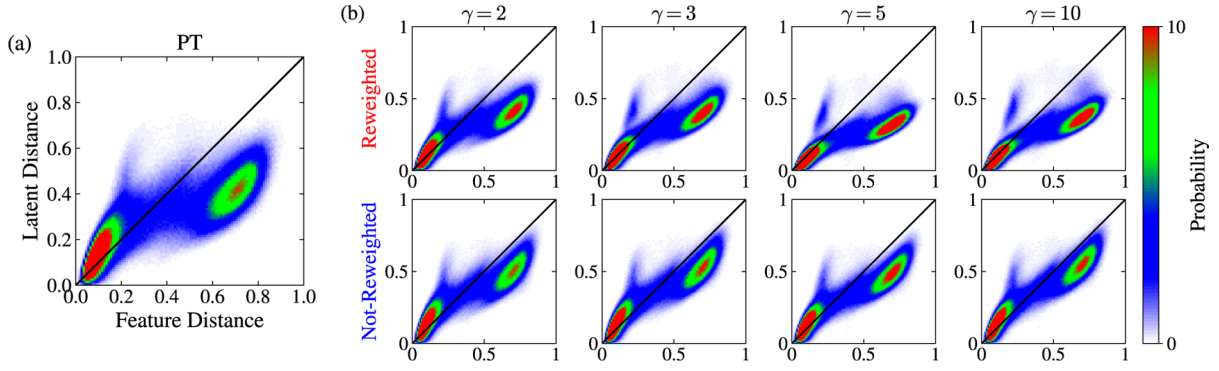


Figure 12. Results for alanine dipeptide in vacuum at 300 K. The joint probability density functions for the pairwise distances in the high-dimensional feature space and the low-dimensional latent space for the embeddings shown in Figure 10. We show the results for the (a) PT and (b) WT-MetaD embeddings (evaluated on the PT simulation data). These histograms show the similarities between distances in the feature and latent spaces. For an embedding that preserves distances accurately, the density would lie on the identity line $y = x$ (shown as a black line). We normalize the distances to lie in the range 0 to 1.

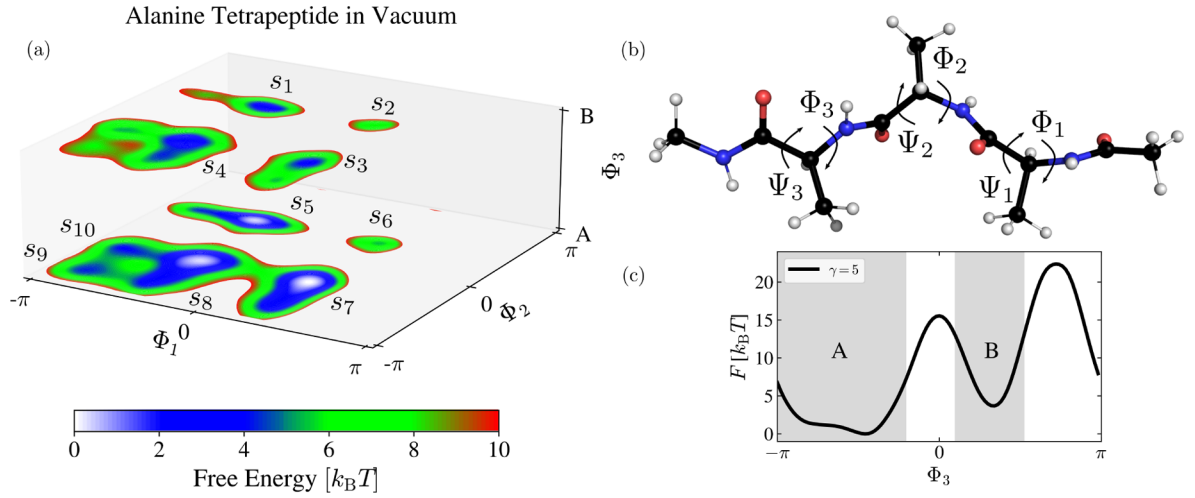


Figure 13. Results for alanine tetrapeptide in vacuum at 300 K. (a) Conditional FESs (eq 21), obtained from the WT-MetaD simulation, shown as a function of Φ_1 and Φ_2 for two minima of Φ_3 labeled as A and B. We denote the ten metastable states as s_1 to s_{10}. (b) Alanine tetrapeptide system with the backbone dihedral angles $\mathbf{\Phi} \equiv (\Phi_1, \Phi_2, \Phi_3)$ and $\mathbf{\Psi} \equiv (\Psi_1, \Psi_2, \Psi_3)$ that we use as the input features for the MRSE embeddings. (c) Free energy profile $F(\Phi_3)$, obtained from the WT-MetaD simulation, with the two minima A and B. The gray-shaded area indicates the areas integrated over in eq 21. The FESs are obtained using KDE as described in Section 3.4.

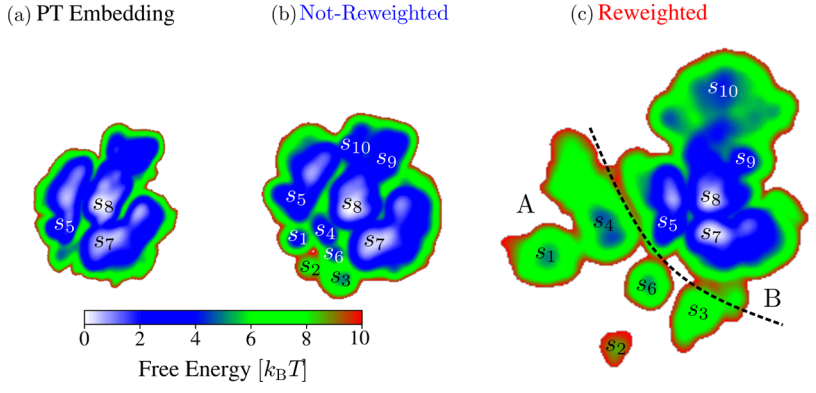


Figure 14. Results for alanine tetrapeptide in vacuum at 300 K. FESs for the MRSE embeddings trained on the unbiased and biased simulation data. (a) PT embedding trained and evaluated on the PT simulation data. (b,c) WT-MetaD embeddings trained and evaluated on the WT-MetaD simulation data. The WT-MetaD embeddings are obtained without (b) and with (c) incorporating weights into the training via a reweighted feature pairwise probability distribution (see eq 9). The FESs are obtained using KDE as described in Section 3.4. The state labels in the FESs correspond to the labeling used in Figure 13a. The embeddings are rescaled so that the equilibrium states are of similar size. The units for the MRSE embeddings are arbitrary and thus not shown.

https://doi.org/10.1021/acs.jpca.1c02869
J. Phys. Chem. A 2021, 125, 6286−6302

$\mathbf{\Phi}$ obtained with different methods (Figure S12 in Supporting Information), and keeping in mind that the 300 K replica from the PT simulation only describes well the lower-lying metastable states, we find that all simulations are in good agreement. Therefore, we conclude that the WT-MetaD and PT simulations are converged.

To show the results from the three-dimensional CV space on a two-dimensional surface, we consider a conditional FES where the landscape is given as a function of Φ_1 and Φ_2 conditioned on values of Φ_3 being in one of the two distinct minima shown in Figure 13c. We label these minima as A and B. We define the conditional FES as

$$F(\Phi_1, \Phi_2|\Phi_3 \in S) = -\frac{1}{\beta}\log \int_S \mathrm{d}\Phi_3\, \mathrm{e}^{-\beta F(\mathbf{\Phi})} \tag{21}$$

where $F(\mathbf{\Phi})$ is the FES obtained from the WT-MetaD simulation (aligned such that its minimum is at zero), S is either the A or B minima, and we integrate over the regions indicated by the gray areas in Figure 13c. We show the two conditional FESs in Figure 13a. Through a visual inspection of Figure 13, we can identify ten different metastable states, denoted as s_1 to s_{10}. Three of the states, s_5, s_7, and s_8, are sampled properly in the 300 K replica of the PT simulation, and thus, we consider them as the equilibrium metastable states. The rest of the metastable states are located higher in free energy and only sampled accurately in the WT-MetaD simulation. The number of the metastable states observed in Figure 13a is in agreement with a recent study of Giberti et al.[53]

We can judge the quality of the MRSE embeddings based on whether they can correctly capture the metastable states in only two dimensions. As input features for the MRSE embeddings, we use sines and cosines of backbone dihedral angles $\mathbf{\Phi}$ and $\mathbf{\Psi}$ (12 features in total), instead of heavy atom distances as we do in the previous section for alanine dipeptide. We use weight-tempered random sampling with $\alpha = 2$ to select landmarks for the training of the WT-MetaD embeddings.

We show the PT and WT-MetaD embeddings in Figure 14. We can see that the PT embedding in Figure 14a is able to accurately describe the equilibrium metastable states (i.e., s_5, s_7, and s_8). However, as expected, the PT embedding cannot describe all ten metastable states, as the 300 K replica in the PT simulation rarely samples the higher-lying states.

In contrast, we can see that the WT-MetaD embeddings in Figure 14b,c capture accurately all ten metastable states. By visual inspection of the simulation data, we can assign state labels for the embeddings in Figure 14, corresponding to the states labeled in Figure 13a. One interesting aspect of the MRSE embeddings in Figure 14 is that they similarly map the equilibrium states, even if we obtain the embeddings from different simulation data sets (PT and WT-MetaD). This similarity underlines the consistency of our approach. The fact that both the reweighted and not-reweighted WT-MetaD embeddings capture all ten states suggests we could use both embeddings as CVs for biasing.

However, we can observe that the reweighted embedding has a better visual separation of the states. For example, we can see this for the separation between s_9 and s_{10}. Furthermore, we can see that the reweighted embedding separates the states from the A and B regions better than the not-reweighted embedding. In the reweighted embedding, states s_1 to s_4 are close to each other and separated from states s_5–s_{10} as indicated by line drawn in Figure 14c. Therefore, we can conclude that the reweighted WT-MetaD embedding is of better quality and better represents distances between metastable states for this system. These results show that we need to employ a reweighted feature pairwise probability distribution for more complex systems.

5. DISCUSSION AND CONCLUSIONS

We present MRSE, a general framework that unifies enhanced sampling and ML for constructing CVs. MRSE builds on top of ideas from SNE methods.[24−26,39] We introduce several advancements to SNE methods that make MRSE suitable for constructing CVs from biased data obtained from enhanced sampling simulations.

We show that this method can construct CVs automatically by learning mapping from a high-dimensional feature space to a low-dimensional latent space via a deep NN. We can use the trained NN to project any given point in feature space to CV space without rerunning the training procedure. Furthermore, we can obtain the derivatives of the learned CVs with respect to the input features and bias the CVs within an enhanced sampling simulation. In future work, we will use this property by employing MRSE within an enhanced sampling scheme where the CVs are iteratively improved.[33,34,37]

In this work, we focus entirely on the training of the embeddings, using training data sets obtained from both unbiased simulation and biased simulation employing different biasing strengths (i.e., bias factors in WT-MetaD). As the "garbage in, garbage out" adage applies to ML (a model is only as good as training data), to eliminate the influence of incomplete sampling, we employ idealistic sampling conditions that are not always achievable in practice.[40] In future work, we will need to consider how MRSE performs under less ideal sampling conditions. One possible option to address this issue is to generate multiple embeddings by running independent training attempts and score them using the maximum caliber principle, as suggested in ref 40.

The choice of the input features depends on the physical system under study. In this work, we use conventional features, that is, microscopic coordinates, distances, and dihedral angles, as they are a natural choice for the model systems considered here. In general, the features can be complicated functions of the microscopic coordinates.[19] For example, symmetry functions have been used as input features in studies of phase transformations in crystalline systems.[17,18] Additionally, features may be correlated or simply redundant. See ref 103 for a general outline of feature selection in unsupervised learning. We will explore the usage of more intricate input features and modern feature selection methods[104,105] for MRSE embeddings in future work.

One of the issues with using kernel-based dimensionality reduction methods, such as diffusion maps[23] or SNE methods,[24] is that the user needs to select the bandwidths (i.e., the scale parameters $\boldsymbol{\varepsilon}$) when using the Gaussian kernels. In t-SNE,[25,26] the Gaussian bandwidths are optimized by fitting to a parameter called perplexity. We can view the perplexity as the effective number of neighbors in a manifold.[25,26] However, this only redirects the issue as the user still needs to select the perplexity parameter.[106] Larger perplexity values lead to a larger number of nearest neighbors and an embedding less sensitive to small topographic structures in the data. Conversely, lower perplexity values lead to fewer neighbors and ignore global information in favor of the local environment. However, what if several length scales characterize the data? In this case, it is impossible to represent the density of the data with a single set of bandwidths,

https://doi.org/10.1021/acs.jpca.1c02869
J. Phys. Chem. A 2021, 125, 6286−6302

so viewing multiple embeddings obtained with different perplexity values is quite common.[106]

In MRSE, we circumvent the issue of selecting the Gaussian bandwidths or the perplexity value by employing a multiscale representation of feature space. Instead of a single Gaussian kernel, we use a Gaussian mixture where each term has its bandwidths optimized for a different perplexity value. We perform this procedure in an automated way by employing a range of perplexity values representing several length scales. This mixture representation allows describing both the local and global characteristics of the underlying data topography. The multiscale nature of MRSE makes the method particularly suitable for tackling complex systems, where the free energy landscape consists of several metastable states of different sizes and shapes. However, as we have seen in Section 4.3, also model systems may exhibit such complex behavior.

Employing nonlinear dimensionality reduction methods is particularly problematic when considering training data obtained from enhanced sampling simulations. In this case, the feature samples are drawn from a biased probability distribution, and each feature sample carries a statistical weight that we need to take into account. In MRSE, we take the weights into account when selecting the representative feature samples (i.e., landmarks) for the training. For this, we introduce a weight-tempered selection scheme that allows us to obtain landmarks that strike a balance between equilibrium distribution and capturing important metastable states lying higher in free energy. This weight-tempered random sampling method depends on a tempering parameter α that allows us to tune between obtaining equilibrium and biased distribution of landmarks. This parameter is case-dependent and similar in spirit to the bias factor γ in WT-MetaD. Generally, α should be selected so that every crucial metastable state is densely populated. However, α should not be too large, as it may result in including feature samples from unimportant higher-lying free energy regions.

The weight-tempered random sampling algorithm is inspired by and bears a close resemblance to the WT-FPS landmark selection algorithm, introduced by Ceriotti et al.[73] For small values of the tempering parameter α, both methods give similar results, as discussed in Section 4.2. The main difference between the algorithms lies in the limit $\alpha \to \infty$. In weight-tempered random sampling, we obtain a landmark distribution that is the same as the biased distribution from the enhanced sampling simulation. On the other hand, WT-FPS results in landmarks that are sampled uniformly distributed from the simulation data set. Due to usage of FPS[107] in the initial stage, WT-FPS is computationally more expensive. Thus, as we are interested in a landmark selection obtained using smaller values of α and do not want uniformly distributed landmarks, we prefer weight-tempered random sampling.

The landmarks obtained with weight-tempered random sampling still carry statistical weights that can vary considerably. Thus, we also incorporate the weights into the training by employing a reweighted feature pairwise probability distribution. To test the effect of this reweighting, we constructed MRSE embeddings without including the weights in the training. Then, we only take the weights into account during the landmark selection. For alanine dipeptide, the reweighted MRSE embeddings are more consistent and slightly better than the not-reweighted ones. For the more challenging alanine tetrapeptide case, both the reweighted and not-reweighted embeddings capture all the metastable states. However, we can observe that the reweighted embedding has a better visual separation of states. Thus, we can conclude from these two systems that employing a reweighted feature pairwise probability distribution is beneficial or even essential, especially when considering more complex systems. Nevertheless, this is an issue that we need to consider further in future work.

Finally, we have implemented the MRSE method and weight-tempered random sampling in the open-source PLUMED library for enhanced sampling and free energy computation.[9,60] Having MRSE integrated into PLUMED is of significant advantage. We can use MRSE with the most popular MD codes and learn CVs in postprocessing and on the fly during a molecular simulation. Furthermore, we can employ the learned CVs with the various CV-based enhanced sampling methods implemented in PLUMED. We will make our code publicly available under an open-source license by contributing it as a module called LowLearner to the official PLUMED repository in the future. In the meantime, we release an initial implementation of LowLearner with our data. The archive of our data is openly available at Zenodo[81] (DOI: 10.5281/zenodo.4756093). PLUMED input files and scripts required to replicate the results are available from the PLUMED NEST[60] under plumID:21.023 at https://www.plumed-nest.org/eggs/21/023/.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jpca.1c02869.

Entropy of the reweighted feature pairwise probability distribution; KL divergence loss for a full set of training data; description of WT-FPS; effective landmark CV distribution for weight-tempered random sampling; details about the clustering used in Figure 7; bandwidth values for KDE; loss function learning curves; additional embeddings for the Müller-Brown potential; feature preprocessing in alanine dipeptide system; alanine dipeptide embeddings for different values of α in weight-tempered random sampling; alanine dipeptide embeddings for $\alpha = 2$ in WT-FPS; alanine dipeptide embeddings for different random seed values; and convergence of alanine tetrapeptide simulations (PDF)

■ AUTHOR INFORMATION

Corresponding Authors

Jakub Rydzewski − *Institute of Physics, Faculty of Physics, Astronomy and Informatics, Nicolaus Copernicus University, 87-100 Torun, Poland;* orcid.org/0000-0003-4325-4177; Email: jr@fizyka.umk.pl

Omar Valsson − *Max Planck Institute for Polymer Research, Mainz D-55128, Germany;* orcid.org/0000-0001-7971-4767; Email: valsson@mpip-mainz.mpg.de

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jpca.1c02869

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

We want to thank Ming Chen (UC Berkeley) and Gareth Tribello (Queen's University Belfast) for valuable discussions and Robinson Cortes-Huerto, Oleksandra Kukharenko, and Joseph F. Rudzinski (Max Planck Institute for Polymer

Research) for carefully reading over an initial draft of the manuscript. J.R. gratefully acknowledges financial support from the Foundation for Polish Science (FNP). We acknowledge using the MPCDF (Max Planck Computing & Data Facility) DataShare.

REFERENCES

(1) Abrams, C.; Bussi, G. Enhanced sampling in molecular dynamics using metadynamics, replica-exchange, and temperature-acceleration. *Entropy* **2014**, *16*, 163−199.
(2) Valsson, O.; Tiwary, P.; Parrinello, M. Enhancing important fluctuations: Rare events and metadynamics from a conceptual viewpoint. *Ann. Rev. Phys. Chem.* **2016**, *67*, 159−184.
(3) Yang, Y. I.; Shao, Q.; Zhang, J.; Yang, L.; Gao, Y. Q. Enhanced sampling in molecular dynamics. *J. Chem. Phys.* **2019**, *151*, 070902.
(4) Bussi, G.; Laio, A. Using metadynamics to explore complex free-energy landscapes. *Nat. Rev. Phys.* **2020**, *2*, 200−212.
(5) Noé, F.; Clementi, C. Collective variables for the study of long-time kinetics from molecular trajectories: Theory and methods. *Curr. Opin. Struct. Biol.* **2017**, *43*, 141−147.
(6) Pietrucci, F. Strategies for the exploration of free energy landscapes: Unity in diversity and challenges ahead. *Rev. Phys.* **2017**, *2*, 32−45.
(7) Rydzewski, J.; Nowak, W. Ligand diffusion in proteins via enhanced sampling in molecular dynamics. *Phys. Life Rev.* **2017**, *22*−*23*, 58−74.
(8) Fiorin, G.; Klein, M. L.; Hénin, J. Using collective variables to drive molecular dynamics simulations. *Mol. Phys.* **2013**, *111*, 3345−3362.
(9) Tribello, G. A.; Bonomi, M.; Branduardi, D.; Camilloni, C.; Bussi, G. PLUMED 2: New feathers for an old bird. *Comput. Phys. Commun.* **2014**, *185*, 604−613.
(10) Sidky, H.; Colón, Y. J.; Helfferich, J.; Sikora, B. J.; Bezik, C.; Chu, W.; Giberti, F.; Guo, A. Z.; Jiang, X.; Lequieu, J.; et al. SSAGES: Software suite for advanced general ensemble simulations. *J. Chem. Phys.* **2018**, *148*, 044104.
(11) Murdoch, W. J.; Singh, C.; Kumbier, K.; Abbasi-Asl, R.; Yu, B. Definitions, methods, and applications in interpretable machine learning. *Proc. Natl. Acad. Sci. U.S.A.* **2019**, *116*, 22071−22080.
(12) Xie, J.; Gao, R.; Nijkamp, E.; Zhu, S.-C.; Wu, Y. N. Representation learning: A statistical perspective. *Annu. Rev. Stat. Appl.* **2020**, *7*, 303−335.
(13) Wang, Y.; Lamim Ribeiro, J. M.; Tiwary, P. Machine learning approaches for analyzing and enhancing molecular dynamics simulations. *Curr. Opin. Struct. Biol.* **2020**, *61*, 139−145.
(14) Noé, F.; Tkatchenko, A.; Müller, K.-R.; Clementi, C. Machine learning for molecular simulation. *Ann. Rev. Phys. Chem.* **2020**, *71*, 361−390.
(15) Gkeka, P.; Stoltz, G.; Barati Farimani, A.; Belkacemi, Z.; Ceriotti, M.; Chodera, J. D.; Dinner, A. R.; Ferguson, A. L.; Maillet, J.-B.; Minoux, H.; et al. Machine learning force fields and coarse-grained variables in molecular dynamics: Application to materials and biological systems. *J. Chem. Theory Comput.* **2020**, *16*, 4757−4775.
(16) Sidky, H.; Chen, W.; Ferguson, A. L. Machine learning for collective variable discovery and enhanced sampling in biomolecular simulation. *Mol. Phys.* **2020**, *118*, No. e1737742.
(17) Geiger, P.; Dellago, C. Neural networks for local structure detection in polymorphic systems. *J. Chem. Phys.* **2013**, *139*, 164105.
(18) Rogal, J.; Schneider, E.; Tuckerman, M. E. Neural-network-based path collective variables for enhanced sampling of phase transformations. *Phys. Rev. Lett.* **2019**, *123*, 245701.
(19) Musil, F.; Grisafi, A.; Bartók, A. P.; Ortner, C.; Csányi, G.; Ceriotti, M. Physics-inspired structural representations for molecules and materials. **2021**, arXiv:2101.04673. arXiv preprint.
(20) Coifman, R. R.; Lafon, S.; Lee, A. B.; Maggioni, M.; Nadler, B.; Warner, F.; Zucker, S. W. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proc. Natl. Acad. Sci. U.S.A.* **2005**, *102*, 7426−7431.
(21) Coifman, R. R.; Lafon, S. Diffusion maps. *Appl. Comput. Harmon. Anal.* **2006**, *21*, 5−30.
(22) Nadler, B.; Lafon, S.; Coifman, R. R.; Kevrekidis, I. G. Diffusion maps, spectral clustering and reaction coordinates of dynamical systems. *Appl. Comput. Harmon. Anal.* **2006**, *21*, 113−127.
(23) Coifman, R. R.; Kevrekidis, I. G.; Lafon, S.; Maggioni, M.; Nadler, B. Diffusion maps, reduction coordinates, and low dimensional representation of stochastic systems. *Multiscale Model. Simul.* **2008**, *7*, 842−864.
(24) Hinton, G.; Roweis, S. T. Stochastic neighbor embedding. *Neural Inf. Process. Syst.* **2002**, *15*, 833−840.
(25) van der Maaten, L.; Hinton, G. Visualizing data using *t*-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579−2605.
(26) van der Maaten, L. Learning a parametric embedding by preserving local structure. *J. Mach. Learn. Res.* **2009**, *5*, 384−391.
(27) Ceriotti, M.; Tribello, G. A.; Parrinello, M. Simplifying the representation of complex free-energy landscapes using sketch-map. *Proc. Natl. Acad. Sci. U.S.A.* **2011**, *108*, 13023−13028.
(28) Tribello, G. A.; Ceriotti, M.; Parrinello, M. Using sketch-map coordinates to analyze and bias molecular dynamics simulations. *Proc. Natl. Acad. Sci. U.S.A.* **2012**, *109*, 5196−5201.
(29) McInnes, L.; Healy, J.; Melville, J. UMAP: Uniform manifold approximation and projection for dimension reduction. **2018**, arXiv:1802.03426. arXiv preprint.
(30) Ma, A.; Dinner, A. R. Automatic method for identifying reaction coordinates in complex systems. *J. Phys. Chem. B* **2005**, *109*, 6769−6779.
(31) Chen, W.; Ferguson, A. L. Molecular enhanced sampling with autoencoders: On-the-fly collective variable discovery and accelerated free energy landscape exploration. *J. Comput. Chem.* **2018**, *39*, 2079−2102.
(32) Hernández, C. X.; Wayment-Steele, H. K.; Sultan, M. M.; Husic, B. E.; Pande, V. S. Variational encoding of complex dynamics. *Phys. Rev. E* **2018**, *97*, 062412.
(33) Ribeiro, J. M. L.; Bravo, P.; Wang, Y.; Tiwary, P. Reweighted autoencoded variational Bayes for enhanced sampling (RAVE). *J. Chem. Phys.* **2018**, *149*, 072301.
(34) Chen, W.; Tan, A. R.; Ferguson, A. L. Collective variable discovery and enhanced sampling using autoencoders: Innovations in network architecture and error function design. *J. Chem. Phys.* **2018**, *149*, 072312.
(35) Wehmeyer, C.; Noé, F. Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics. *J. Chem. Phys.* **2018**, *148*, 241703.
(36) Li, S.-H.; Dong, C.-X.; Zhang, L.; Wang, L. Neural canonical transformation with symplectic flows. *Phys. Rev. X* **2020**, *10*, 021020.
(37) Zhang, J.; Chen, M. Unfolding hidden barriers by active enhanced sampling. *Phys. Rev. Lett.* **2018**, *121*, 010601.
(38) Lemke, T.; Peter, C. EncoderMap: Dimensionality reduction and generation of molecule conformations. *J. Chem. Theory Comput.* **2019**, *15*, 1209−1215.
(39) van der Maaten, L. Accelerating *t*-SNE using tree-based algorithms. *J. Mach. Learn. Res.* **2014**, *15*, 3221−3245.
(40) Pant, S.; Smith, Z.; Wang, Y.; Tajkhorshid, E.; Tiwary, P. Confronting pitfalls of AI-augmented molecular dynamics using statistical physics. *J. Chem. Phys.* **2020**, *153*, 234118.
(41) Torrie, G. M.; Valleau, J. P. Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling. *J. Comput. Phys.* **1977**, *23*, 187−199.
(42) Kästner, J. Umbrella sampling. *Wiley Interdiscip. Rev.: Comput. Mol. Sci.* **2011**, *1*, 932−942.
(43) Darve, E.; Pohorille, A. Calculating free energies using average force. *J. Chem. Phys.* **2001**, *115*, 9169.
(44) Comer, J.; Gumbart, J. C.; Hénin, J.; Lelièvre, T.; Pohorille, A.; Chipot, C. The adaptive biasing force method: Everything you always wanted to know but were afraid To ask. *J. Phys. Chem. B* **2015**, *119*, 1129−1151.

https://doi.org/10.1021/acs.jpca.1c02869
J. Phys. Chem. A 2021, 125, 6286−6302

(45) Lesage, A.; Lelièvre, T.; Stoltz, G.; Hénin, J. Smoothed biasing forces yield unbiased free energies with the extended-system adaptive biasing force method. *J. Phys. Chem. B* **2016**, *121*, 3676−3685.
(46) Maragakis, P.; van der Vaart, A.; Karplus, M. Gaussian-mixture umbrella sampling. *J. Phys. Chem. B* **2009**, *113*, 4664−4673.
(47) Laio, A.; Parrinello, M. Escaping free-energy minima. *Proc. Natl. Acad. Sci. U.S.A.* **2002**, *99*, 12562−12566.
(48) Barducci, A.; Bussi, G.; Parrinello, M. Well-tempered metadynamics: A smoothly converging and tunable free-energy method. *Phys. Rev. Lett.* **2008**, *100*, 020603.
(49) Valsson, O.; Parrinello, M. Variational approach to enhanced sampling and free energy calculations. *Phys. Rev. Lett.* **2014**, *113*, 090601.
(50) Valsson, O.; Parrinello, M. *Handbook of materials modeling: Methods: Theory and modeling*; Andreoni, W., Yip, S., Eds.; Springer International Publishing: Cham, 2020, pp 621−634.
(51) Invernizzi, M.; Parrinello, M. Rethinking metadynamics: From bias potentials to probability distributions. *J. Phys. Chem. Lett.* **2020**, *11*, 2731−2736.
(52) Invernizzi, M.; Piaggi, P. M.; Parrinello, M. Unified approach to enhanced sampling. *Phys. Rev. X* **2020**, *10*, 041034.
(53) Giberti, F.; Tribello, G. A.; Ceriotti, M. Global free energy landscapes as a smoothly joined collection of local maps. *J. Chem. Theory Comput.* **2021**, *17* (6), 3292−3308.
(54) Dama, J. F.; Parrinello, M.; Voth, G. A. Well-tempered metadynamics converges asymptotically. *Phys. Rev. Lett.* **2014**, *112*, 240602.
(55) Tiwary, P.; Parrinello, M. A time-independent free energy estimator for metadynamics. *J. Phys. Chem. B* **2015**, *119*, 736−742.
(56) Bonomi, M.; Barducci, A.; Parrinello, M. Reconstructing the equilibrium Boltzmann distribution from well-tempered metadynamics. *J. Comput. Chem.* **2009**, *30*, 1615−1621.
(57) Branduardi, D.; Bussi, G.; Parrinello, M. Metadynamics with adaptive Gaussians. *J. Chem. Theory Comput.* **2012**, *8*, 2247−2254.
(58) Giberti, F.; Cheng, B.; Tribello, G. A.; Ceriotti, M. Iterative unbiasing of quasi-equilibrium sampling. *J. Chem. Theory Comput.* **2019**, *16*, 100−107.
(59) Schäfer, T. M.; Settanni, G. Data reweighting in metadynamics simulations. *J. Chem. Theory Comput.* **2020**, *16*, 2042−2052.
(60) PLUMED Consortium. Promoting transparency and reproducibility in enhanced molecular simulations. *Nat. Methods* **2019**, *16*, 670−673. https://www.plumed-nest.org/consortium.html
(61) Rydzewski, J.; Nowak, W. Machine learning based dimensionality reduction facilitates ligand diffusion paths assessment: A case of cytochrome P450cam. *J. Chem. Theory Comput.* **2016**, *12*, 2110−2120.
(62) Zhou, H.; Wang, F.; Tao, P. *t*-distributed stochastic neighbor embedding method with the least information loss for macromolecular simulations. *J. Chem. Theory Comput.* **2018**, *14*, 5499−5510.
(63) Spiwok, V.; Kříž, P. Time-lagged *t*-distributed stochastic neighbor embedding (*t*-SNE) of molecular simulation trajectories. *Front. Mol. Biosci.* **2020**, *7*, 132.
(64) Fleetwood, O.; Carlsson, J.; Delemotte, L. Identification of ligand-specific G-protein coupled receptor states and prediction of downstream efficacy via data-driven modeling. *eLife* **2021**, *10*, No. e60715.
(65) Globerson, A.; Chechik, G.; Pereira, F.; Tishby, N. Euclidean embedding of co-occurrence data. *J. Mach. Learn. Res.* **2007**, *8*, 2265−2295.
(66) Cover, T. M.; Thomas, J. A. *Elements of information theory*, 2nd ed.; John Wiley & Sons, 2006.
(67) Lee, J. A.; Peluffo-Ordóñez, D. H.; Verleysen, M. Multiscale stochastic neighbor embedding: Towards parameter-free dimensionality reduction. *European Symposium on Artificial Neural Networks*, 2014.
(68) De Bodt, C.; Mulders, D.; Verleysen, M.; Lee, J. A.. Perplexity-free *t*-SNE and twice Student *tt*-SNE. *European Symposium on Artificial Neural Networks*, 2018.
(69) Crecchi, F.; de Bodt, C.; Verleysen, M.; Lee, J. A.; Bacciu, D. Perplexity-free parametric *t*-SNE. **2020**, arXiv:2010.01359. arXiv preprint.
(70) Sammon, J. W. A Nonlinear Mapping for Data Structure Analysis. *IEEE Trans. Comput.* **1969**, *C-18*, 401−409.
(71) Marimont, R. B.; Shapiro, M. B. Nearest neighbour searches and the curse of dimensionality. *IMA J. Appl. Math.* **1979**, *24*, 59−70.
(72) Kullback, S.; Leibler, R. A. On information and sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79−86.
(73) Ceriotti, M.; Tribello, G. A.; Parrinello, M. Demonstrating the transferability and the descriptive power of sketch-map. *J. Chem. Theory Comput.* **2013**, *9*, 1521−1532.
(74) Long, A. W.; Ferguson, A. L. Landmark diffusion maps (L-dMaps): Accelerated manifold learning out-of-sample extension. *Appl. Comput. Harmon. Anal.* **2019**, *47*, 190−211.
(75) Tribello, G. A.; Gasparotto, P. Using dimensionality reduction to analyze protein trajectories. *Front. Mol. Biosci.* **2019**, *6*, 46.
(76) Tribello, G. A.; Gasparotto, P. *Biomolecular Simulations: Methods and protocols*; Springer, 2019, p 453. DOI: 10.1007/978-1-4939-9608-7̇19.
(77) Swendsen, R. H.; Wang, J.-S. Replica Monte Carlo simulation of spin-glasses. *Phys. Rev. Lett.* **1986**, *57*, 2607.
(78) Koblents, E.; Míguez, J. A population Monte Carlo scheme with transformed weights and its application to stochastic kinetic models. *Stat. Comput.* **2015**, *25*, 407−425.
(79) Bortz, A. B.; Kalos, M. H.; Lebowitz, J. L. A new algorithm for Monte Carlo simulation of Ising spin systems. *J. Comput. Phys.* **1975**, *17*, 10−18.
(80) Gil-Ley, A.; Bussi, G. Enhanced conformational sampling using replica exchange with collective-variable tempering. *J. Chem. Theory Comput.* **2015**, *11*, 1077−1085.
(81) Rydzewski, J.; Valsson, O. *Multiscale Reweighted Stochastic Embedding (MRSE): Deep Learning of Collective Variables for Enhanced Sampling*, Version 1.0.0 [Data set]. 2021; 10.5281/zenodo.4756093.
(82) Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An imperative style, high-performance deep learning library. *Neural Inf. Process. Syst.* **2019**, *32*, 8024−8035.
(83) Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; Lerer, A. Automatic differentiation in PyTorch. *Neural Inf. Process. Syst.* **2017**, *31*, 1−4.
(84) Müller, K.; Brown, L. D. Location of saddle points and minimum energy paths by a constrained simplex optimization procedure. *Theor. Chim. Acta* **1979**, *53*, 75−93.
(85) Bussi, G.; Parrinello, M. Accurate sampling using Langevin dynamics. *Phys. Rev. E* **2007**, *75*, 056707.
(86) Abraham, M. J.; Murtola, T.; Schulz, R.; Páll, S.; Smith, J. C.; Hess, B.; Lindahl, E. GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX* **2015**, *1-2*, 19−25.
(87) Hornak, V.; Abel, R.; Okur, A.; Strockbine, B.; Roitberg, A.; Simmerling, C. Comparison of multiple Amber force fields and development of improved protein backbone parameters. *Proteins* **2006**, *65*, 712−725.
(88) Bussi, G.; Donadio, D.; Parrinello, M. Canonical sampling through velocity rescaling. *J. Chem. Phys.* **2007**, *126*, 014101.
(89) Hess, B. P-LINCS: A parallel linear constraint solver for molecular simulation. *J. Chem. Theory Comput.* **2008**, *4*, 116−122.
(90) Hinton, G. E.; Salakhutdinow, R. R. Reducing the dimensionality of data with neural networks. *Science* **2006**, *313*, 504−507.
(91) Maas, A. L.; Hannun, A. Y.; Ng, A. Y. Rectifier nonlinearities improve neural network acoustic models. *J. Mach. Learn. Res.* **2013**, *30*, 3.
(92) Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929−1958.
(93) Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. *J. Mach. Learn. Res.* **2010**, *9*, 249−256.

(94) Kingma, D. P.; Ba, J. Adam: A method for stochastic optimization. *3rd International Conference for Learning Representations*, 2015; Vol. *3*.
(95) Reddi, S. J.; Kale, S.; Kumar, S. On the convergence of Adam and beyond. **2019**, arXiv:1904.09237. arXiv preprint.
(96) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825−2830.
(97) Valsson, O.; Parrinello, M. Well-tempered variational approach to enhanced sampling. *J. Chem. Theory Comput.* **2015**, *11*, 1996−2002.
(98) Tiwary, P.; Berne, B. J. Spectral gap optimization of order parameters for sampling complex molecular systems. *Proc. Natl. Acad. Sci. U.S.A.* **2016**, *113*, 2839.
(99) McCarty, J.; Parrinello, M. A variational conformational dynamics approach to the selection of collective variables in metadynamics. *J. Chem. Phys.* **2017**, *147*, 204109.
(100) Yang, Y. I.; Parrinello, M. Refining collective coordinates and improving free energy representation in variational enhanced sampling. *J. Chem. Theory Comput.* **2018**, *14*, 2889−2894.
(101) Bonati, L.; Zhang, Y.-Y.; Parrinello, M. Neural networks-based variationally enhanced sampling. *Proc. Natl. Acad. Sci. U.S.A.* **2019**, *116*, 17641−17647.
(102) Piana, S.; Laio, A. A bias-exchange approach to protein folding. *J. Phys. Chem. B* **2007**, *111*, 4553−4559.
(103) Dy, J. G.; Brodley, C. E. Feature selection for unsupervised learning. *J. Mach. Learn. Res.* **2004**, *5*, 845−889.
(104) Ravindra, P.; Smith, Z.; Tiwary, P. Automatic mutual information noise omission (AMINO): generating order parameters for molecular systems. *Mol. Syst. Des. Eng.* **2020**, *5*, 339−348.
(105) Cersonsky, R. K.; Helfrecht, B. A.; Engel, E. A.; Ceriotti, M. Improving sample and feature selection with principal covariates regression. **2020**, arXiv:2012.12253. arXiv preprint.
(106) Wattenberg, M.; Viégas, F.; Johnson, I. How to use *t*-SNE effectively. *Distill* **2016**, *1*, No. e2. https://distill.pub/2016/misread-tsne/
(107) Hochbaum, D. S.; Shmoys, D. B. A best possible heuristic for the *k*-center problem. *Math. Oper. Res.* **1985**, *10*, 180−184.

NOTE ADDED AFTER ASAP PUBLICATION

This paper was published on July 2, 2021. Due to production error, some of the equations in the Methods and Results sections were rendered incorrectly. The corrected version was reposted on July 7, 2021.

https://doi.org/10.1021/acs.jpca.1c02869
J. Phys. Chem. A 2021, 125, 6286−6302