Systematic Control of Collective Variables Learned from Variational Autoencoders

Jacob I. Monroe[1] and Vincent K. Shen[1]
Chemical Sciences Division, National Institute of Standards and Technology, Gaithersburg, Maryland 20899-8320, USA

(*Electronic mail: vincent.shen@nist.gov)

(*Electronic mail: jacob.monroe@nist.gov)

(Dated: 2 August 2022)

Variational autoencoders (VAEs) are rapidly gaining popularity within molecular simulation for discovering low-dimensional, or latent, representations, which are critical for both analyzing and accelerating simulations. However, it remains unclear how the information a VAE learns is connected to its probabilistic structure, and, in turn, its loss function. Previous studies have focused on feature engineering, *ad hoc* modifications to loss functions, or adjustment of the prior to enforce desirable latent space properties. By applying effectively arbitrarily flexible priors via normalizing flows, we focus instead on how adjusting the structure of the decoding model impacts the learned latent coordinate. We systematically adjust the power and flexibility of the decoding distribution, observing that this has a significant impact on the structure of the latent space as measured by a suite of metrics developed in this work. By also varying weights on separate terms within each VAE loss function, we show that the level of detail encoded can be further tuned. This provides practical guidance for utilizing VAEs to extract varying resolutions of low-dimensional information from molecular dynamics and Monte Carlo simulations.

I. INTRODUCTION

Identifying low-dimensional representations for interpreting the results of molecular simulations, or for efficiently biasing simulations to better explore configuration space, remains a significant challenge in the field. Autoencoding structures have rapidly gained popularity for learning low-dimensional collective variables (CVs), or latent spaces, from molecular simulation data[1–11]. In particular, variational autoencoders (VAEs)[12] provide a Bayesian modeling framework that can tune the properties of a learned latent space through the form of the probabilistic models composing the VAE[13]. However, systematic studies of the impact of probabilistic modeling assumptions on the learned latent space, especially those associated with the form of the decoding distribution, are, to our knowledge, nonexistent in the published scientific literature. To address this, we propose metrics to quantify these impacts and examine their behavior as we progressively adjust the modeling power of decoding models in VAEs with highly flexible priors that make minimal assumptions on the form of the latent space.

Choosing a probabilistic decoding model directly sets the training objective, or loss function for a VAE, in turn critically impacting the CVs that are learned through model training. VAEs consist of an encoding-decoding structure in which neural networks taking full-space $\mathbf{x}$ or latent space $\mathbf{z}$ coordinates are trained to predict encoding $q(\mathbf{z}|\mathbf{x})$ and decoding $P(\mathbf{x}|\mathbf{z})$ probability distributions. The model is also defined in terms of a prior distribution $P(\mathbf{z})$ that can remain fixed, or, as we allow here, can be learned during training. The full VAE loss $\mathcal{L}_{\text{total}}$ is an upper bound on the negative log likelihood of a Bayesian model for the probability density of the training data[12]

$$\mathcal{L}_{\text{total}} \geq \left\langle -\ln \frac{P(\mathbf{x}|\mathbf{z})\,P(\mathbf{z})}{P(\mathbf{z}|\mathbf{x})} \right\rangle_{\mathcal{P}(\mathbf{x})} \tag{1}$$

The ensemble average is taken over the training data distribution $\mathcal{P}(\mathbf{x})$. $\mathcal{L}_{\text{total}}$ is an upper bound due to the introduction of the variational distribution $q(\mathbf{z}|\mathbf{x})$ to approximate $P(\mathbf{z}|\mathbf{x})$[12]

$$\begin{aligned}\mathcal{L}_{\text{total}} &= \langle -\ln P(\mathbf{x}) + \text{KL}\left(q(\mathbf{z}|\mathbf{x}), P(\mathbf{z}|\mathbf{x})\right)\rangle_{\mathcal{P}(\mathbf{x})} &(2)\\ &= \left\langle \langle -\ln P(\mathbf{x}|\mathbf{z})\rangle_{q(\mathbf{z}|\mathbf{x})} + \beta_{\text{reg}}\text{KL}\left(q(\mathbf{z}|\mathbf{x}), P(\mathbf{z})\right)\right\rangle_{\mathcal{P}(\mathbf{x})} &(3)\\ &= \mathcal{L}_{\text{recon}} + \beta_{\text{reg}}\mathcal{L}_{\text{KL}} &(4)\end{aligned}$$

The second term on the right of Eq. 3 is the Kullback-Leibler (KL) divergence[14] between the encoding distribution $q(\mathbf{z}|\mathbf{x})$ and the prior $P(\mathbf{z})$, with β_{reg} defined as the regularization weight[15]. For $\beta_{\text{reg}} = 1$ we recover the exact variational upper bound while other values adjust the weight between the reconstruction and KL divergence loss terms indicated in Eq. 4, which, more specifically, are

$$\mathcal{L}_{\text{recon}} = \left\langle \langle -\ln P(\mathbf{x}|\mathbf{z})\rangle_{q(\mathbf{z}|\mathbf{x})} \right\rangle_{\mathcal{P}(\mathbf{x})} \tag{5}$$

$$\mathcal{L}_{\text{KL}} = \langle \text{KL}\left(q(\mathbf{z}|\mathbf{x}), P(\mathbf{z})\right)\rangle_{\mathcal{P}(\mathbf{x})} \tag{6}$$

If β_{reg} is set to zero, the loss is identical to that of a standard autoencoder, though a VAE still differs in that its encoding and decoding involve stochastic sampling. In this case, however, it is expected that the encoding and decoding distributions approach delta functions, approaching the behavior of a standard autoencoder.

Eq. 5 makes it clear that the selected decoding model expresses our assumptions concerning the system we are modeling. In molecular simulation, this directly involves the physical nature of interactions between degrees of freedom — do we assume that atomic coordinates are independent of each other or probabilistically coupled? Just as the form of the prior $P(\mathbf{z})$ places constraints on the properties of the latent

space distribution, the decoding distribution $P(\mathbf{x}|\mathbf{z})$ limits the types of interactions that may be captured in the full configurational space. However, the properties of the learned latent space will directly depend on the form of the decoding model because a VAE is driven by Eq. 3 to encode information that is useful for increasing the probability of inputs under $P(\mathbf{x}|\mathbf{z})$. Latent information can be used to enhance the probabilistic modeling power of the decoding distribution, lowering the reconstruction loss. A price is paid, however, through an increase in the KL loss term. The KL loss effectively requires that the latent space ignore extraneous information, focusing only on that which benefits and improves the decoding model. As we will show, this can mean that the learned latent space contains no useful information if the decoding model is either too weak or too powerful.

Ideally, we would like to always select a VAE model to learn the most meaningful latent space, which first requires defining a "good" latent coordinate. There are two primary schools of thought on determining CVs from simulations: either pick those associated with the slowest timescales in the system, or those that traverse the most volume in configurational space[1]. While a number of studies have explored the use of time-lagged autoencoders or VAEs to identify slow degrees of freedom[8,9,16,17], we focus here on using VAEs to identify CVs that best describe equilibrium configuration space, which applies to both molecular dynamics simulations that have a sense of time and Monte Carlo (MC) simulations that do not. To be useful, it is desirable that such CVs be: 1) interpretable by humans, 2) smooth and differentiable mappings from full-space coordinates, and 3) maximally expressive in that they can flexibly learn to capture low-dimensional representations regardless of the specific details of the system being simulated.

The first desirable property of latent coordinates is that they be interpretable by a human, and in that sense meaningful. In image processing, there has similarly been a considerable amount of interest in using VAEs to determine "disentangled" representations[15,18–20], which enables interpretation of highly non-linear encodings. Locatello *et al.*[21] have shown, however, that identifying disentangled latent spaces is largely a matter of luck, depending heavily on the random number seed for starting the training run. The reason is simply that, to the VAE, any linear combination of the most informative latent coordinates is equally optimal, meaning that human interpretability is never guaranteed without human intervention in constructing the latent space. Tiwary and coworkers have proposed a promising avenue for interpreting learned latent spaces by also learning how to represent these coordinates in terms of linear combinations of human-proposed CVs that best approximate the non-linear encoding of the VAE[4,22]. This technique is always available as a post-processing tool for interpreting the results of training a VAE, but does not impact the properties of the learned latent space.

While standard autoencoders and VAEs naturally identify mappings to CVs that are differentiable, and hence can be used to apply biasing forces to molecular dynamics simulations, it is also advantageous for them to be smooth, avoiding large discontinuities in latent space. Such jumps in CV values are problematic when enhancing sampling of configuration space by applying biasing potentials or forces along a latent coordinate. This means that related configurations should be projected to similar locations in latent space, which has motivated the addition of distance-preserving terms to the loss function of standard autoencoders (i.e., only the reconstruction term)[5,23]. Applied to a VAE, this could be thought of as an added restraint on the form of the encoding distribution, but we argue this is unnecessary due to the stochastic nature of the encoding. The encoder of a VAE maps a full-space configuration to an entire region of latent space rather than to a single point, with any unnecessary separation of similar configurations in latent space penalized by the KL loss. Chen, Tan, and Ferguson[3] also propose modifications to a standard autoencoder loss, but by introducing multiple reconstruction objectives (e.g., atomic coordinates and distances between sets of atoms). This is a form of feature engineering and effectively places different weights on full-space degrees of freedom, or relationships between them, that the decoder attempts to reproduce, which those authors show has an impact on the quality of the learned CVs for enhancing sampling. Multiple objectives or different choices of inputs and outputs will not change the underlying decoder model in a VAE, but will change the features that are emphasized. We will not focus on feature engineering in this work other than to demonstrate that the choice of coordinate system for VAE inputs and outputs modifies the difficulty of the modeling task and should therefore influence choices of structure and complexity in a decoding model.

Discovered latent spaces should also ideally be maximally expressive, free from unintentional constraints placed upon their structure. For instance, Chen, Tan, and Ferguson[3] demonstrated that, without explicitly enabling latent space periodicity, autoencoders have difficulty identifying low-dimensional representations associated with dihedral angles. Gaussian mixture encoders and priors have been introduced recently to increase the modeling capacity of $P(\mathbf{z})$[10,11]. This is particularly appropriate for learning clusterings of data, with configurations subdivided into different Gaussians within latent space. In this work, we apply even more flexible priors based on normalizing flows[13,24,25]. A normalizing flow learns to transform one random variable following a known probability density into another that satisfies the probability density of the data used for training[26]. In training our VAE models with a flow on the prior, we learn effectively arbitrarily complex priors $P(\mathbf{z})$. Unlike in a Gaussian mixture with a fixed functional form and number of components, a prior based on a flow is free to take on any continuous form and include as many peaks as necessary for encoding information useful to the decoder.

For a flow defining the transformation $\mathbf{z} = f(\mathbf{z}')$ from a random variable $\mathbf{z}'$ satisfying a normal distribution with zero mean and unit variance $\mathcal{N}(\mathbf{z}';\mathbf{0},\mathbf{1})$ to our latent space variable $\mathbf{z}$, the KL loss term becomes

$$\mathcal{L}_{\mathrm{KL}} = \left\langle \ln q(\mathbf{z}|\mathbf{x}) - \ln \mathcal{N}\left(f^{-1}(\mathbf{z});\mathbf{0},\mathbf{1}\right) - \ln R_{\mathbf{z}\rightarrow\mathbf{z}'}(\mathbf{z}) \right\rangle_{q(\mathbf{z}|\mathbf{x})} \tag{7}$$

$R_{\mathbf{z}\rightarrow\mathbf{z}'}(\mathbf{z})$ is the absolute value of the Jacobian determinant of

the flow transformation $\mathbf{z}' = f^{-1}(\mathbf{z})$, which is readily calculated for the transformations we implement (see Section II). With $P(\mathbf{z})$ fixed to a simple distribution that is easy to sample from, like a standard normal, the expressivity of the encoding is restrained by the KL divergence term, which may result in an overly simple, uninformative latent space and a poor approximation of $q(\mathbf{z}|\mathbf{x})$ to $P(\mathbf{z}|\mathbf{x})$[24,25,27]. A flow relaxes restraints on the form of the learned latent distribution but, through Eq. 7, still penalizes highly structured, complex latent spaces that are difficult to transform to a standard normal. A prior based on a flow is also preferred when generating new configurations because it avoids a scenario in which the distribution of encoded values fails to match the chosen prior $P(\mathbf{z})$ and is thus unknown and difficult to sample from directly. While use of a flow on the prior may thus generally improve VAE models, our emphasis here is that it enables near-maximally expressive and flexible latent spaces so that the impact of the decoder model alone may be assessed.

To understand the fundamental impact of the ideas presented above, such as the interplay between regularization strength and decoder flexibility, we consider a variety of toy and simple molecular systems. For each, we train a number of VAE models in replicate to assess variations across training runs. Systematic adjustments are made to the power of the decoding model, as well as to the regularization weight β_{reg} that adjusts the balance between reconstruction and KL loss terms. We compare learned latent spaces across different decoders and values of β_{reg} via a number of novel metrics for assessing latent space structure and utilization. Our results reveal that the form of the probabilistic decoding model and weight on the KL loss significantly alter the latent space, or even if any information is encoded. Based on our observations, we map out conditions under which a meaningful latent space is expected to be learned and provide practical guidance in using VAEs to learn low-dimensional representations of molecular systems.

II. METHODS

A. VAE Models and Training

VAEs consist of an encoding-decoding structure in which neural networks taking full-space $\mathbf{x}$ or latent space $\mathbf{z}$ coordinates are trained to predict encoding $q(\mathbf{z}|\mathbf{x})$ and decoding $P(\mathbf{x}|\mathbf{z})$ probability distributions. The model is also defined in terms of a prior distribution $P(\mathbf{z})$ that can remain fixed, or, as we allow here, can be learned during training. As is typical, we set the functional form of the encoding distribution to independent Gaussians along each latent dimension, with means and log-variances output by the last layer of the encoder neural network. Encoder neural networks consist of two fully-connected hidden layers of 50 units each for all 2D systems and 300 units each for molecular systems. Hidden layers receive rectified linear unit (ReLU) activation while we apply no activation to output layers. We transform periodic degrees of freedom (e.g., dihedral angles) to sine-cosine pairs before passing through the encoder network. Schematics describing the architectures of all components of our VAE models are provided in Figs. S12-S20 of the Supplementary Material (SM) and code may be found at https://github.com/JIMonroe/vae-mc.

To focus on the impact of decoding model assumptions on learned latent spaces, throughout this work we use extremely flexible priors based on normalizing flows, with a VAE structure as described by Chen *et al.*[13]. A normalizing flow defines a bijective mapping based on neural networks that transforms a simple, analytically known distribution into the unknown distribution of the training data[26]. We use neural spline flows[28] with a RealNVP structure[29] as implemented in TensorFlow Probability[30] to learn a transformation $\mathbf{z}' = f^{-1}(\mathbf{z})$ to turn latent encodings $\mathbf{z}$ into samples from a random variable $\mathbf{z}'$ that satisfies a standard normal distribution. All flows consist of 4 RealNVP blocks, each transforming half of the dimensions by using the untransformed dimensions as inputs to neural networks that output parameters for a monotonically increasing rational quadratic spline associated with that block (Fig. S19). For flows on a single dimension, constants are input to the neural networks defining splines so that a fixed transformation independent of the input data is learned for each block. We define a domain of $[-10,10]$ for our spline transformations with 31 spline knot locations and slopes learned for all transformed dimensions by a fully connected neural network with a single hidden dimension of 20 units and ReLU activation.

With fixed choices for the encoding and prior probability distributions, we systematically vary the representational power of the decoder probability distribution. Fig. 1 schematically illustrates the decoder models that we employ, with each option leading to a different form for the reconstruction loss. Further details of decoder architectures may be found in Figs. S14-S18. In the simplest case (Fig. 1a), the decoder neural network only predicts the means of full-space degrees of freedom $\boldsymbol{\mu}(\mathbf{z})$, assuming independent Gaussian distributions with variances fixed at unity $\boldsymbol{\sigma}^2(\mathbf{z}) = \mathbf{1}$. Using this specific Gaussian probability density for the full-space coordinates in the decoder probability distribution leads to the classic mean-squared error reconstruction loss and so we label it the "MSE" decoder model.

$$-\ln P_{\text{MSE}}(\mathbf{x}|\mathbf{z}) = \frac{1}{2}\sum_{i}^{M}\left[(x_i - \mu_i(\mathbf{z}))^2 + \ln 2\pi\right] \tag{8}$$

The summation in Eq. 8 runs over all M fine-grained degrees of freedom. The $\ln 2\pi$ term comes from the proper normalization constant of a Gaussian distribution with unit variance. While such a constant will not affect training, it is explicitly represented here to emphasize that the form of the loss function is a direct consequence of choosing a decoder probabilistic model. Notably, the MSE model assumes that, given $\mathbf{z}$, all elements of $\mathbf{x}$ are probabilistically independent of each other, which is called conditional independence, i.e.,

$$P(\mathbf{x}|\mathbf{z}) = \prod_i P(x_i|\mathbf{z}) \tag{9}$$

As such, we term the next decoder model "conditionally independent," since it maintains this probabilistic structure but

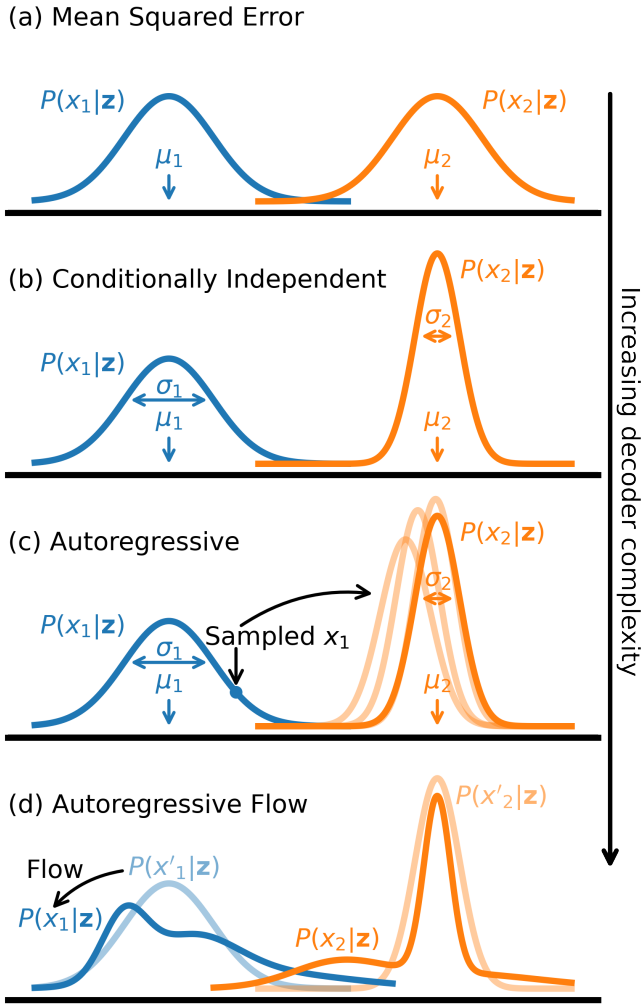

FIG. 1. Four different decoding models are applied within this work: (a) conditionally independent normal distributions with means determined by the decoder neural network and all variances set to 1, which gives rise to the well-known mean-squared error reconstruction loss, (b) conditionally independent normal distributions with both means and variances determined by the decoder neural network, (c) autoregressive normal distributions, where the means and variances of a given degree of freedom depend on previously sampled degrees of freedom, and (d) a decoder applying an autoregressive normalizing flow to samples drawn from independent normal distributions determined by the decoder network, both adding correlations between degrees of freedom and removing assumptions of normality.

now allows the variance of decoder normal distributions to be determined by the decoder neural network.

$$-\ln P_{\text{CondInd}}(\mathbf{x}|\mathbf{z}) = \frac{1}{2}\sum_i^M \left[\frac{(x_k - \mu_i(\mathbf{z}))^2}{\sigma_i^2(\mathbf{z})} + \ln 2\pi + \ln \sigma_i^2(\mathbf{z}) \right] \tag{10}$$

In Eq. 10, the variances are elements in a vector since all off-diagonal elements of the covariance matrix are zeros, reiterating the conditional independence of the decoder probability distributions. While it has been proposed to introduce correlations into decoder models by predicting full covariance matrices[31,32], we instead introduce correlations through an autoregressive decoding probabilistic structure[13,33], i.e.,

$$P(\mathbf{x}|\mathbf{z}) = \prod_i P(x_i|\mathbf{z}, \mathbf{x}_{<i}) \tag{11}$$

In "autoregressive" decoder models, we sample each degree of freedom in turn with a fixed ordering, allowing sampled values to influence the means and variances of degrees of freedom yet to be sampled. While this does not modify the form of the log-probability in Eq. 10, it drastically changes the functional dependencies of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$.

$$-\ln P_{\text{Auto}}(\mathbf{x}|\mathbf{z}) = \frac{1}{2}\sum_i^M \left[\frac{(x_i - \mu_i(\mathbf{z}, \mathbf{x}_{<i}))^2}{\sigma_i^2(\mathbf{z}, \mathbf{x}_{<i})} + \ln 2\pi + \ln \sigma_i^2(\mathbf{z}, \mathbf{x}_{<i}) \right] \tag{12}$$

In the most powerful decoder, termed our "flow" model, we relax the assumption of normality in our decoding distributions by applying an autoregressive normalizing flow that learns a bijective transformation $\mathbf{x} = f(\mathbf{x}')$. This leads to a reconstruction loss of

$$-\ln P_{\text{Flow}}(\mathbf{x}|\mathbf{z}) = \frac{1}{2}\sum_i^M \left[\frac{\left(f^{-1}(\mathbf{x})_i - \mu_i(\mathbf{z})\right)^2}{\sigma_i^2(\mathbf{z})} + \ln 2\pi + \ln \sigma_i^2(\mathbf{z}) \right] - \ln R_{\mathbf{x}\to\mathbf{x}'}(\mathbf{x}) \tag{13}$$

During generation, the decoder neural network first predicts means and variances of conditionally independent normal distributions in the space of $\mathbf{x}' = f^{-1}(\mathbf{x})$. Samples drawn from these distributions are transformed by the autoregressive flow into samples on $\mathbf{x}$. We write Eq. 13 in terms of $\mathbf{x}$ and transformations to $\mathbf{x}'$ since during training we evaluate the likelihood of data under this model and must implement the inverse flow transformation to compute the model probability.

All decoder models contain fully-connected hidden layers utilizing tanh activations with 50 units for 2D systems and 300 units for molecular systems. MSE and conditionally independent models contain 2 hidden layers before outputting means and log-variances with no activation. Autoregressive decoders follow the same architecture, but then use masked neural networks, as in Ref.[34], to predict shifts in the means and log-variances due to correlations between degrees of freedom. Latent space values are provided directly to the masked neural networks as conditional inputs to encourage use of latent information, as suggested by the success of skip connections[35]. Flow decoders only contain a single hidden layer and produce conditionally independent means and log-variances of normal distributions without activation in their next layer. We use TensorFlow Probability[30] to implement two stacked masked autoregressive flows[36] with neural spline transformations[28] as

the final layer in our flow decoders. As in autoregressive decoders, $\mathbf{z}$ values are provided as conditional inputs for the flow transformations. For autoregressive and flow decoders, we use von Mises rather than Gaussian distributions for periodic degrees of freedom, which provides an additional advantage over MSE and conditionally independent decoders.

To implement all models, we use TensorFlow[37] (version 2.4.0) and TensorFlow Probability[30]. We train models in batches of 200 samples with the Adam optimizer[38], setting the learning rate to 0.0001 and other parameters to their default values in TensorFlow. For systems in 2D, models train for 100 epochs at each value of β_{reg}, while models for molecular systems are train for 300 epochs. Models with $\beta_{\text{reg}} \geq 1$ are trained in turn, carrying over parameters to the next value of β_{reg} and annealing from the last value of β_{reg} over 20 epochs for 2D systems and 40 epochs for molecular systems. For $\beta_{\text{reg}} = 1$, annealing from 0 starts with randomly generated parameters rather than those of the model trained for the full number of epochs at $\beta_{\text{reg}} = 0$. Slowly annealing from one β_{reg} to another helps prevent the the collapse of the latent space to uninformative noise before the VAE has time to identify a more structured encoding[39].

B. Latent Metrics

We have developed a suite of novel metrics for characterizing latent space properties for any system and VAE model. The average Jeffreys Divergence (JD)[40] between encoding distributions characterizes the degree to which configurations are distinguished within the latent space. JD is defined as the sum of two KL divergences with the reference distribution switched, though here we write it as an average

$$\begin{aligned}\text{JD}_{\text{encode}} = \frac{1}{2}\big(&\text{KL}\left(q\left(\mathbf{z}|\mathbf{x}_i\right), q\left(\mathbf{z}|\mathbf{x}_j\right)\right) \\ &+\text{KL}\left(q\left(\mathbf{z}|\mathbf{x}_j\right), q\left(\mathbf{z}|\mathbf{x}_i\right)\right)\big)\end{aligned} \tag{14}$$

Each KL divergence is between encoding distributions defined by data samples $\mathbf{x}_j$ and $\mathbf{x}_i$, with the average $\text{JD}_{\text{encode}}$ determined by averaging over all pairs (or in practice all pairs within a batch) of the training data. While Eq. 14 is general for any choice of encoding distribution, for normal distributions with diagonal covariance structure, as implemented here, the KL divergence is

$$\begin{aligned}\text{KL}\left(q\left(\mathbf{z}|\mathbf{x}_i\right), q\left(\mathbf{z}|\mathbf{x}_j\right)\right) = \frac{1}{2}\sum_k^{D_l}\Bigg(&\frac{\sigma_{i,k}^2}{\sigma_{j,k}^2} + \frac{\left(\mu_{j,k}-\mu_{i,k}\right)^2}{\sigma_{j,k}^2} \\ &-1+\ln\frac{\sigma_{j,k}^2}{\sigma_{i,k}^2}\Bigg)\end{aligned} \tag{15}$$

The summation runs over the dimensionality of the latent space D_l with $\mu_{i,k}$ and $\sigma_{i,k}^2$ being the mean and variance of the normal distribution along latent dimension k associated with the i^{th} training data point. Large values of the average $\text{JD}_{\text{encode}}$ identify memorization of data, a form of overfitting where each training data point is encoded in its own volume in latent space and encoding distributions have almost no overlap. Small values approaching zero indicate an unused latent space where all encoding distributions overlap fully and the VAE does not distinguish between configurations in the latent space. Computing this quantity in a dimension-wise fashion differentiates latent dimensions based on their utilization.

To estimate the relative complexity of the latent distribution, or roughly how structured it is compared to Gaussian noise with the same mean $\mathcal{N}\left(\mathbf{z};\boldsymbol{\mu}_{\mathbf{z}},\mathbf{1}\right)$, we define

$$\begin{aligned}\text{JD}_{\text{prior}} = \frac{1}{2}\big(&\text{KL}\left(q\left(\mathbf{z}\right), \mathcal{N}\left(\mathbf{z};\boldsymbol{\mu}_{\mathbf{z}},\mathbf{1}\right)\right) \\ &+\text{KL}\left(\mathcal{N}\left(\mathbf{z};\boldsymbol{\mu}_{\mathbf{z}},\mathbf{1}\right), q\left(\mathbf{z}\right)\right)\big)\end{aligned} \tag{16}$$

Note that Eq. 16 uses the variational prior under the training data distribution $q\left(\mathbf{z}\right) = \langle q\left(\mathbf{z}|\mathbf{x}\right)\rangle_{\mathcal{P}(\mathbf{x})}$ rather than the prior learned based on a normalizing flow $P\left(\mathbf{z}\right)$. Though these should be similar to satisfy the training objective in Eq. 7, we draw $\mathbf{z}$ samples from the former by sequentially drawing from the training data distribution $\mathcal{P}\left(\mathbf{x}\right)$ followed by the encoding distribution $q\left(\mathbf{z}|\mathbf{x}\right)$ and from the latter by drawing $\mathbf{z}'$ samples from a standard normal and passing through the flow $f\left(\mathbf{z}'\right)$. JD_{prior} will be small when $q\left(\mathbf{z}\right)$ matches the prior in the coordinate transformed by the flow $\mathbf{z}'$, indicating that the Jacobian determinant term in Eq. 7 approaches zero. This can happen, however, when $\mathbf{z}'$ and $\mathbf{z}$ differ only by an arbitrary additive shift, and so we compare to a normal with the same mean of $\boldsymbol{\mu}_{\mathbf{z}} = \left\langle \langle \mathbf{z} \rangle_{q(\mathbf{z}|\mathbf{x})} \right\rangle_{\mathcal{P}(\mathbf{x})}$ but unit variance, which would be the simplest prior one of our VAE models could learn. To calculate JD_{prior}, we approximate the KL divergences in Eq. 16 as

$$\begin{aligned}\text{KL}\left(q\left(\mathbf{z}\right), \mathcal{N}\left(\mathbf{z};\boldsymbol{\mu}_{\mathbf{z}},\mathbf{1}\right)\right) \approx \Big\langle \big\langle &\ln\mathcal{N}\left(f^{-1}\left(\mathbf{z}\right);\mathbf{0},\mathbf{1}\right) \\ &+\ln R_{\mathbf{z}\to\mathbf{z}'}\left(\mathbf{z}\right) \\ &-\ln\mathcal{N}\left(\mathbf{z};\boldsymbol{\mu}_{\mathbf{z}},\mathbf{1}\right)\big\rangle_{q(\mathbf{z}|\mathbf{x})}\Big\rangle_{\mathcal{P}(\mathbf{x})}\end{aligned} \tag{17}$$

$$\begin{aligned}\text{KL}\left(\mathcal{N}\left(\mathbf{z};\boldsymbol{\mu}_{\mathbf{z}},\mathbf{1}\right), q\left(\mathbf{z}\right)\right) \approx &-S_{\mathcal{N}} \\ &-\big\langle \ln\mathcal{N}\left(f^{-1}\left(\mathbf{z}\right);\mathbf{0},\mathbf{1}\right) \\ &+\ln R_{\mathbf{z}\to\mathbf{z}'}\left(\mathbf{z}\right)\big\rangle_{\mathcal{N}(\mathbf{z};\boldsymbol{\mu}_{\mathbf{z}},\mathbf{1})}\end{aligned} \tag{18}$$

In Eqs. 17 and 18, we have used the learned prior $P\left(\mathbf{z}\right)$ defined by a flow to approximate the probability density of $q\left(\mathbf{z}\right)$, though samples in Eq. 17 are drawn from the variational prior under the training data distribution. In Eq. 18, $S_{\mathcal{N}} = \frac{D_l}{2}\left(\ln\left(2\pi\right)+1\right)$ is the entropy of a standard normal distribution. Note that both JD_{prior} and the average $\text{JD}_{\text{encode}}$ must be small to achieve a small KL loss in Eq. 7. For example, the learned $P\left(\mathbf{z}\right)$ might exactly follow a standard normal distribution, driving JD_{prior} to zero, but each encoding distribution $q\left(\mathbf{z}|\mathbf{x}\right)$ might be sharply peaked, resulting in high average $\text{JD}_{\text{encode}}$. These metrics measure different aspects of the encoding, with average $\text{JD}_{\text{encode}}$ and JD_{prior} separately assessing local and global structure of the learned latent space.

We measure the probabilistic independence of latent dimensions through the total correlation. While we can also assess linear independence through the Pearson correlation matrix, the total correlation identifies whether or not each latent dimension encodes unique information and is defined as

$$\begin{aligned} Q_{\mathrm{TC}} &= \int q(\mathbf{z}) \ln \frac{q(\mathbf{z})}{\prod_k^{D_l} q(z_k)} \mathrm{d}\mathbf{z} \\ &= -S_{q(\mathbf{z})} + \sum_k^{D_l} S_{q(z_k)} \end{aligned} \tag{19}$$

In the second line, $S_{q(\mathbf{z})}$ is the entropy of the full variational distribution under the training data and the sum in the second term is over the entropies of the marginal distributions $S_{q(z_k)}$ of all D_l latent dimensions. Chen *et al.* [18] approximate Q_{TC} through an asymptotically biased Monte Carlo method that provides a loose lower bound. Locatello *et al.* [21] instead apply a Gaussian approximation to estimate the total correlation, analytically calculating Q_{TC} from the mean and covariance matrix of $q(\mathbf{z})$. Since we have learned a normalizing flow defining $P(\mathbf{z})$ that closely approximates $q(\mathbf{z})$, we use this to define a tight lower bound estimate of the total correlation. Specifically, an upper bound on the entropy of the full distribution in Eq. 19 can be estimated by using $P(\mathbf{z})$ to estimate the probability density of $q(\mathbf{z})$.

$$\begin{aligned} S_{q(\mathbf{z})} &\approx \left\langle \left\langle -\ln P(\mathbf{z}) \right\rangle_{q(\mathbf{z}|\mathbf{x})} \right\rangle_{\mathcal{P}(\mathbf{x})} \\ &\approx \left\langle \left\langle -\ln \mathcal{N}\left(f^{-1}(\mathbf{z}); \mathbf{0}, \mathbf{1}\right) - \ln R_{\mathbf{z}\to\mathbf{z}'}(\mathbf{z}) \right\rangle_{q(\mathbf{z}|\mathbf{x})} \right\rangle_{\mathcal{P}(\mathbf{x})} \end{aligned} \tag{20}$$

Note that we expect this upper bound to be tight since Eq. 20 is minimized as part of the KL loss in Eq. 7. Entropies of marginals are estimated through histogramming and numerical integration, which is appropriate when many samples are available for single-dimensional data. High total correlation indicates that different latent dimensions are not probabilistically independent and contain duplicate information.

For VAEs trained on data fully representing the configurational space of a system, the fraction of accepted VAE-based Monte Carlo (MC) moves f_{acc} is a general metric estimating the accuracy with which the probability density of the training data is modeled[41]. In this technique, we use a trained VAE to propose moves that are accepted or rejected according to a Metropolis criterion satisfying detailed balance in the ensemble of the training data — we utilize the potential energy function used to generate the data to define relative configurational probabilities. Unlike VAE loss functions that are typically unbounded and vary with the system and chosen model, f_{acc} is bounded from 0 to 1 and is system and model agnostic. A description and detailed analysis of VAE-based MC moves is presented in our previous publication[41].

C. Model Systems and Simulations

The Mueller potential energy function governs sampling in a 2D space and is defined as in the work of Noé *et al.* [42]. The Deep Boltzmann package[43] is used to evaluate the Mueller potential with TensorFlow[37]. Default parameters are used but the potential is scaled by a factor of 0.1, as in Ref.[42]. To generate samples in the ensemble defined by the Mueller potential, we perform 10 000 MC simulations in parallel using 2D Gaussian displacements with standard deviations set to 0.1. Simulations are run for 100 000 trials and configurations are saved every 4 000 trials after discarding the first 60 000 trials for equilibration, resulting in 100 000 independent samples used for training.

All 2D systems other than the Mueller potential are defined using TensorFlow Probability[30] to create distributions represented by class objects that allow calculation of negative log probabilities (potential energies) and efficient sampling. Our independent unimodal potential is a 2D normal distribution with independent means of (-3, -3) and standard deviations of (1, 1). The independent bimodal distribution consists of identical Gaussian mixtures in each dimension. Independently in each dimension, sampling from a normal distribution with mean -3 and standard deviation of 1 happens with probability 0.6 and another normal distribution with mean of 3 and standard deviation of 1 is selected with probability 0.4. We also consider a correlated bimodal distribution based on the same means and variances just described, but with sampling in the second dimension limited to the same normal distribution randomly selected for the first. The "Beltway" potential consists of two concentric rings of low probability separated by a barrier in Cartesian coordinates. For convenience, we define the potential in polar coordinates via a uniform distribution over $[-\pi, \pi]$ for angles and a 50:50 mixture of two gamma distributions for the radial coordinate, one with concentration (α) of 40 and a rate (β) of 40 and the other with a concentration of 160 and rate of 80. This potential differs in mathematical form from that suggested by Chen, Sidky, and Ferguson [16], but remains similar in its free energy surface and can be sampled efficiently without MC simulations. In all 2D systems defined by TensorFlow Probability distributions, we randomly generate 100 000 independent data points for training.

We use OpenMM[44] to simulate alanine dipeptide (or capped alanine, Ace-Ala-Nme) in vacuum at 300 K with a Langevin thermostat, calculating potential energies with the AMBER99-SB[45] forcefield. Full simulation details are provided in a previous publication[41]. Simulations provide 100 000 independent configurations used for training. All training configurations have rigid translational and rotational motion removed and are randomized in their order. Conversions between Cartesian and bond-angle-torsion (BAT) atomic coordinates are performed using MDAnalysis[46,47]. In training VAEs with BAT coordinates, the translational and rotational reference frame is ignored along with all rigid bonds as these degrees of freedom do not require probabilistic modeling.

III. RESULTS

Fig. 2 reveals how decoder structure and regularization strength both impact the properties of VAE models trained

on Mueller potential data. The Mueller potential itself is shown in Fig. 2a, revealing three basins of unequal depth separated by large barriers along a nonlinear coordinate traversing a narrow valley and two saddles. While simple and low-dimensional, the ability of VAE models to learn a meaningful 1D collective variable for this system depends on both the decoder and regularization strength. Distributions of latent encodings in Fig. 2b commonly exhibit large and small peaks spaced far apart at low regularization strength. The large peak represents the lowest free energy basin while the smaller roughly represents the region of phase space around the second-lowest minimum, overlapping slightly with the nearby highest free energy basin. As β_{reg} increases, the peaks shift closer together and eventually merge into a distribution approximating a standard normal. A normal distribution with zero mean and unit variance minimizes the full KL divergence in Eq. 7 by maximizing the likelihood of the transformed prior and driving the log-determinant term to zero. Note, though, that any constant shift in the mean results in an equally low KL divergence as long as the flow learns this shift, which is why all curves in Fig. 2b have their means shifted to zero for ease of visualization. We only observe a close match to a standard normal in the case of the MSE decoder, with the transition occurring at low β_{reg}. At high enough β_{reg}, however, all distributions except for the autoregressive decoder lose the smaller peak and resemble normal distributions.

Fig. 2c colors input data by the learned latent coordinate, revealing that setting $\beta_{\text{reg}} = 0$ leads all decoder models to learn similar latent spaces that sharply trace out paths passing through saddle points between free energy basins. With increased weight on the KL loss, encodings become less sharp, as indicated by blurrier colorings, and distinctions of points within basins fade. At high enough β_{reg}, the data are partitioned in two, with separation occurring across the highest saddle. As implied by the disappearance of the smaller peak in Fig. 2b, both the least and most complex decoder models (MSE and autoregressive flow, respectively) fail to encode anything other than noise at high enough values of β_{reg}. For an MSE decoder, systematic coloring disappears and no collective variable is identified for $\beta_{\text{reg}} \geq 1$, while this transition occurs around $\beta_{\text{reg}} = 5$ for the flow decoder.

Rather than coloring input data by latent values, we plot and color reconstructions sampled from the probability distributions defined by decoding models in Fig. 2d. It is readily apparent that the MSE decoder, with a fixed variance of 1, is unable to reproduce the data's distribution, even though precise decodings, as well as decoder mean representations, are learned with $\beta_{\text{reg}} = 0$. With the KL loss equally weighted against the reconstruction loss at $\beta_{\text{reg}} = 1$, the MSE decoder simply gives up on any type of useful encoding because the reconstruction is essentially equally poor with or without the latent space. The situation might be improved somewhat by whitening the input data (in this system increasing its variance), but this cannot account for the fact that the local variance of the data changes for a given basin with unique curvature. As a result, whitening may extend the threshold of β_{reg} for which a meaningful latent space is learned, but the reconstructions will undergo only marginal improvement, unless a transformation is employed to somehow locally whiten all free energy basins. A simpler solution is to predict both means and variances of the decoder distributions, as in the conditionally independent model. Reconstructions are significantly improved over the MSE decoder and the regularization strength can be used to tune the features learned in the latent space.

For all decoders more powerful than the MSE, reconstructions are better in higher-probability (lower free energy) regions of phase space, coinciding with sharper coloring that indicates more precise encodings in these areas. With autoregression, and hence correlations between x and y, the shape of the lowest free energy basin is maintained even when regularization strength becomes large enough to prevent precise encodings altogether. Starting at $\beta_{\text{reg}} = 2$ the latent space only divides phase space in two and a conditionally independent Gaussian model is unable to account for the covariance between coordinates around the free energy minimum. With an autoregressive flow, the distribution of the data may be learned without any latent information, which is why reconstructions remain reasonable even after the encoder produces only noise. Over three independent training runs, however, we observe that the autoregressive flow consistently learns structured latent spaces when $\beta_{\text{reg}} < 5$, which we show later depends on the training protocol and the presence of degenerate minima in the loss.

Most of the behaviors discussed in Fig. 2 are summarized more succinctly through the metrics shown in Fig. 3. As explicitly defined in Eq. 14, the average JD between encoding distributions measures the overlap between $q(\mathbf{z}|\mathbf{x}_i)$ and $q(\mathbf{z}|\mathbf{x}_j)$ over many i and j from the training dataset. Large values are consistently observed for $\beta_{\text{reg}} = 0$, indicating that each data point is placed in its own small volume of latent space, or that colorings are sharp in Fig. 2c. Such data memorization is expected with only reconstruction terms in the loss and no regularization effect encouraging minimal usage of latent coordinates. Low average $\text{JD}_{\text{encode}}$ coincides with encoding only noise and no systematic coloring in Fig. 2c. The middle panel of Fig. 3 summarizes how closely a latent distribution $P(z)$ resembles a normal distribution with the same mean but a variance of 1. Corresponding to Fig. 2b, we see that JD_{prior} decreases with increasing regularization strength, indicating that the latent distribution becomes more like a standard normal. Though the average $\text{JD}_{\text{encode}}$ and JD_{prior} are typically correlated, this need not be the case and they each provide useful information for understanding the encoding. While both must be low to obtain a low KL divergence and an unused latent space, JD_{prior} can become small while the average $\text{JD}_{\text{encode}}$ remains large — the latent distribution is then simple and unimodal, but the latent dimension is still being utilized to differentiate data (e.g., Fig. 4). The quality of reconstructions in Fig. 2d are correlated to f_{acc} in Fig. 3. If the fraction of accepted VAE-based MC moves is 1, the VAE model has exactly learned the probability density of the data. Interestingly, f_{acc} is always close to 0 when $\beta_{\text{reg}} = 0$, which shows that this metric distinguishes between memorizing data and learning a probabilistic model for the data. An MSE decoder always has a low f_{acc} for the Mueller potential, while the autoregressive

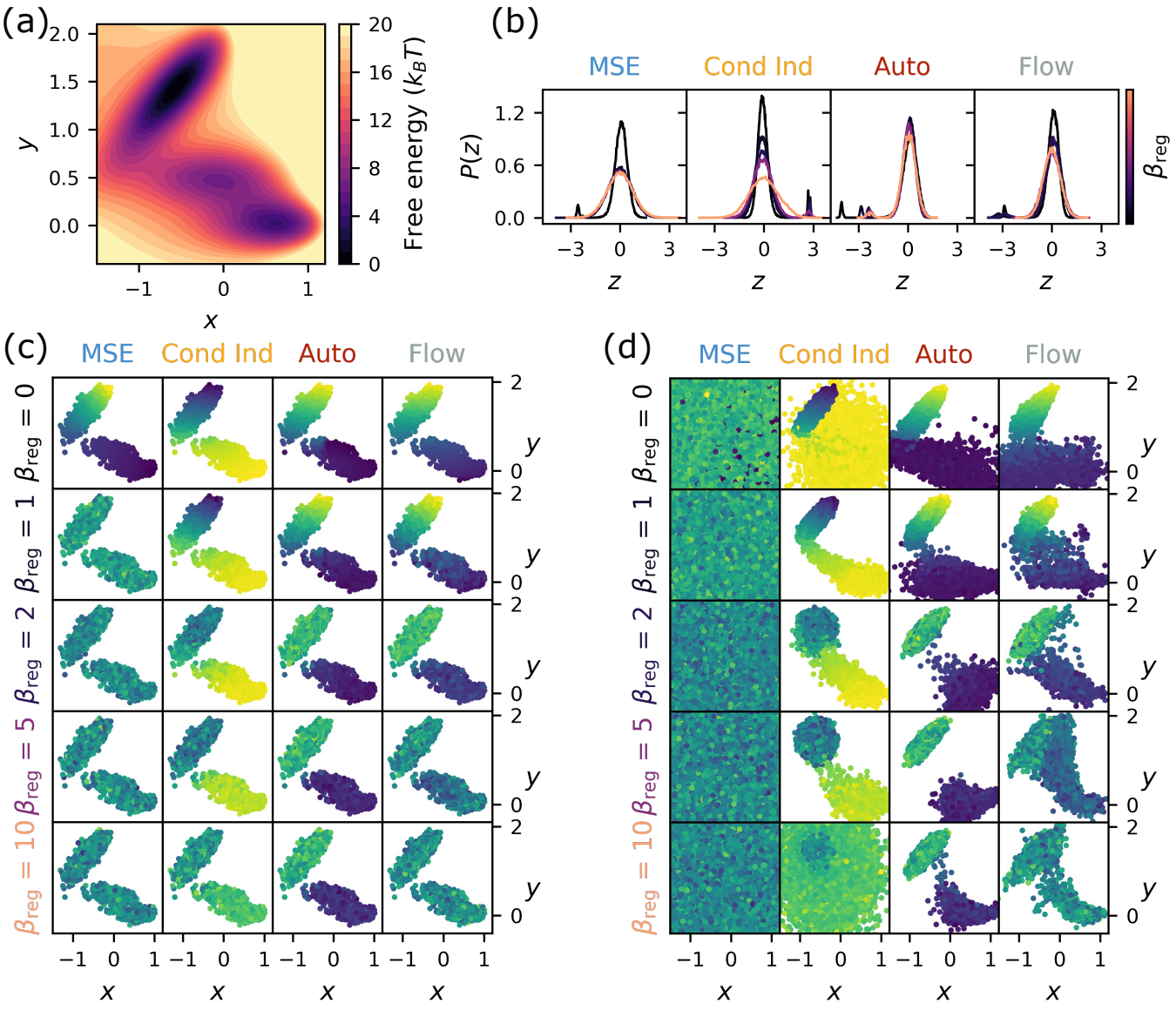

FIG. 2. VAEs trained on data from MC simulations of the (a) 2D Mueller potential learn (b) 1D latent distributions $P(\mathbf{z})$ that depend on both the decoding model and regularization strength β_{reg}. Coordinates of (c) 2D Mueller potential data and (d) reconstructions generated by trained VAE models. Coloring in (c) and (d) is by the value of the latent space coordinate $\mathbf{z}$ drawn from encoding distributions of each input data point (i.e., $q(\mathbf{z}|\mathbf{x}_i)$) in (c) and in turn used to generate the decoding distribution $P(\mathbf{x}_i|\mathbf{z})$ from which the reconstructions in (d) are drawn. Decoder model complexity increases across rows and regularization strength increases moving down columns.

and autoregressive flow decoders maintain high acceptance even with high regularization. When reconstructions begin to break down around $\beta_{\text{reg}} = 2$ for the conditionally independent decoder, f_{acc} correspondingly drops.

We turn to a number of even simpler systems, starting with an independent 2D Gaussian, to better understand the behavior of our proposed latent metrics, as well as to highlight various issues that may arise in identifying latent spaces with VAEs. For a simple 2D Gaussian of unit variance, all decoder models can exactly learn the probability density without usage of the latent space. With no KL loss term, however, all decoders learn structure that is not present in the data, as indicated by sharp coloring of training data (Fig. 4a) and high values of the average $\text{JD}_{\text{encode}}$ (Fig. 4b). This is despite roughly normal latent distributions, as indicated by low values of JD_{prior}. As with the Mueller potential, f_{acc} is close to zero for the highly structured latent spaces at $\beta_{\text{reg}} = 0$, alerting us to the fact that the VAE has not learned an accurate model for the probability density of the training data. With $\beta_{\text{reg}} = 0$, the MSE decoder will approach functioning as a standard autoencoder with a deterministic encoding and mean-squared error loss. This is clear in Fig. 4a, where there is strong similarity between the erroneously learned latent structures of the standard autoencoder and the VAE with an MSE decoder. The encoding distributions of the MSE decoder approach delta functions to mimic a deterministic encoding without the influence

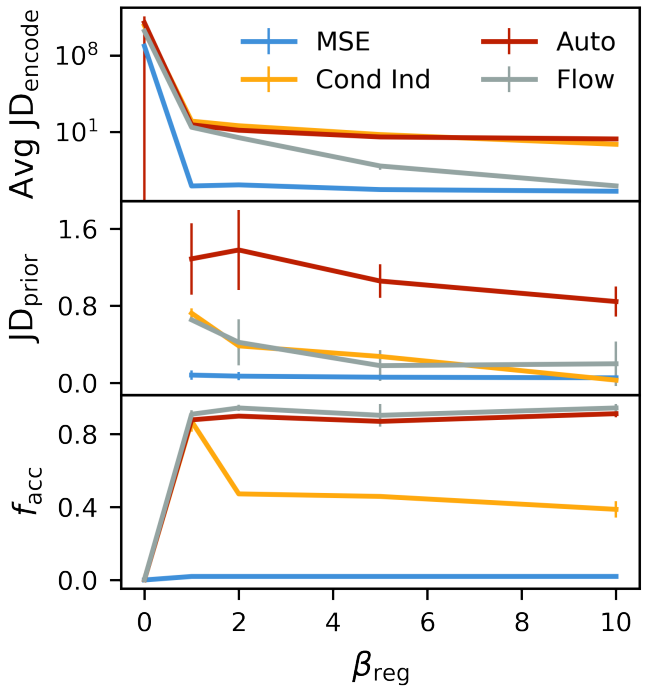

FIG. 3. As a function of regularization strength β_{reg}, we compare latent spaces learned with different decoding models in terms of the average JD between encoding distributions (top), the JD of the learned latent distribution $P(z)$, or prior, from a standard normal with matching mean and unit variance (middle), and the fraction of accepted VAE-based MC moves (bottom). Error bars represent the standard deviation over three independent training runs.

of the KL loss. Interestingly, the MSE decoder is the only model that does not learn some structure in the latent space when $\beta_{\text{reg}} = 1$. Examining loss contributions across independent training runs (Fig. S1), we see that the reconstruction and KL losses tend to exactly compensate each other over a moderate range of values so that the total loss with $\beta_{\text{reg}} = 1$ remains the same. This exposes the presence of degenerate solutions within some range of reconstruction and KL loss values for all models but the MSE decoder, with higher regularization strengths driving the KL term to zero but leaving the total loss unchanged.

A 2D distribution consisting of independent Gaussian mixtures (Fig. 5a) poses a challenge in distinguishing nearly equal free energy basins within the latent space. VAEs with MSE decoders are the only models that consistently learn a latent space partitioned into four separate populations (Fig. S2). Each of the Gaussian distributions in the mixture is of unit variance with the mean shifted. A successful strategy for a decoder is to learn the locations of the means for each basin, encoding this information in latent space, and always setting the variance to 1 for the decoding probability distribution. Since its variance is already fixed to 1, the MSE decoder readily learns a latent space distinguishing the four basins, but similar latent space partitioning poses a difficulty for the more complicated decoder models.

Fig. 5b demonstrates that different training runs arrive at distinct solutions for $\beta_{\text{reg}} = 1$ with different balances of reconstruction and KL divergence loss values with an autoregressive decoder. While variations in the total loss are relatively small (ranging from 4.26 to 4.75), the lowest total loss value of 4.26 (second lowest at 4.28 for three latent peaks) is indeed associated with a learned latent space that contains four distinct populations (run 1 in Fig. 5b-c). Interestingly, f_{acc} is a more sensitive metric of VAE and latent-space quality for this system, displaying larger differences than the total loss with acceptance rates of 0.88 and 0.69 for latent spaces with four and three peaks (runs 1 and 2 in Fig. 5b-c). Due to the highly flexible nature of the prior with a flow, it is possible for any VAE, regardless of decoder model, to learn a partitioning of the free energy basins in latent space, but this is difficult due to the presence of many local minima in the optimization landscape for all but the MSE decoder. Variations in loss values across training runs at β_{reg} values of 1 and 2 for the conditionally independent and autoregressive decoders (Fig. 5b and Fig. S3) provide evidence for these local minima. Both the conditionally independent and autoregressive decoders are theoretically capable of matching the performance of the MSE decoder in Fig. 5d, where f_{acc} is high and the latent space remains utilized and structured (high JD_{prior}) at high regularization strength. However, this is only observed for a single run of the autoregressive decoder, with all other runs falling into local minima that partition the data into two or three peaks in latent space. The flow decoder instead exhibits issues with degenerate minima, as observed in Fig. S3, with different training runs resulting in identical total loss values yet different balances of reconstruction and KL terms at $\beta_{\text{reg}} = 1$. Correspondingly, total losses (Fig. S3) and f_{acc} (Fig. 5d) remain indistinguishable within fluctuations for disparately organized latent spaces in Fig. S2, including for those that are unstructured at higher β_{reg} values. These results point to the fact that a flow decoder is capable of modeling the 2D independent bimodal Gaussian probability density without a latent space, which is also clear from simultaneously high values of f_{acc} and low average $\text{JD}_{\text{encode}}$ for all $\beta_{\text{reg}} \geq 1$ (Fig. 5d). Taken all together, our results for this system indicate that the most meaningful latent space, and the most control over the latent space, will be attained with a decoder model that best matches the physics of the underlying data. This does not necessarily mean the most expressive model should be selected, but rather the decoder that is most effectively constrained in its probabilistic modeling capabilities to best match the underlying probability density to be learned.

Next, we consider a "Beltway" potential similar to that proposed by Chen, Sidky, and Ferguson [16], where the choice of input coordinates significantly changes the difficulty of the modeling task. Though this touches on feature engineering, that is not our focus — as we will show, the choice of coordinate system should also impact which decoder model is selected. Fig. 6a reveals that VAEs trained to learn the distribution of data in Cartesian coordinates tend to learn latent encodings that are mixtures of the radial and angular coordinates. This balance is strongly shifted to angular coordinates for a standard autoencoder with a deterministic encoding and

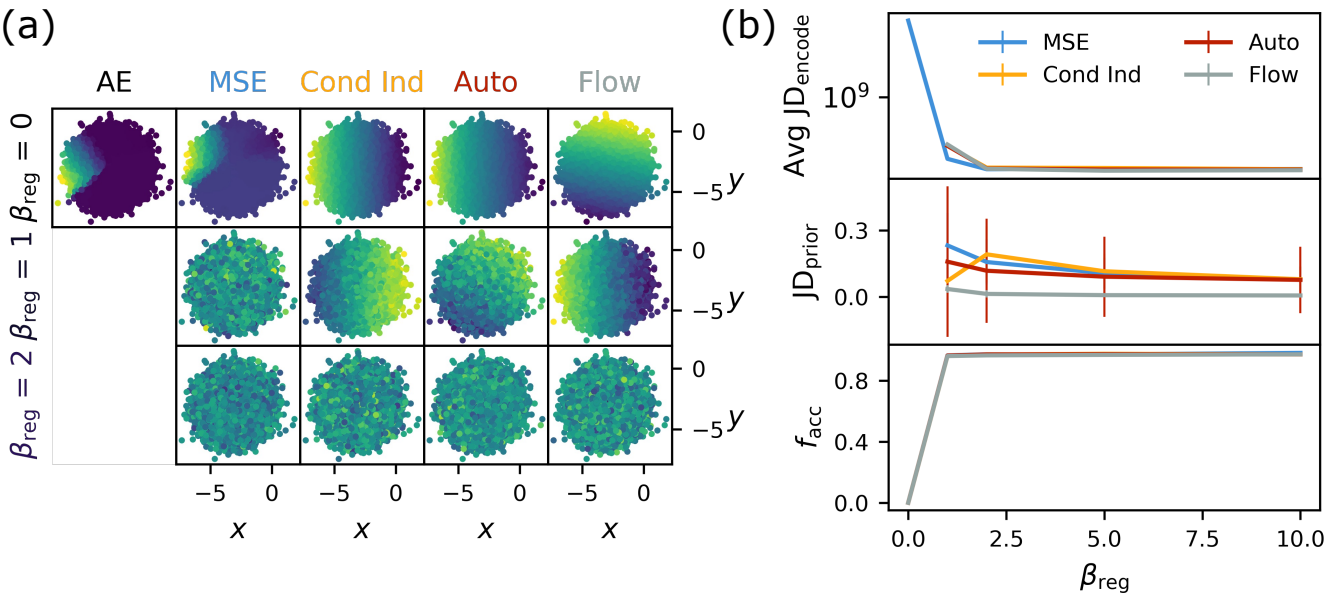

FIG. 4. (a) Data drawn from an independent 2D Gaussian distribution (mean of $(-3,-3)$ and unit variance in both x and y dimensions, as described in Section II C) are colored by encoded 1D latent values for single training runs of various VAE models at differing regularization strengths. AE indicates a standard autoencoder with no variational distribution (a deterministic encoding) and a mean-squared error loss for the decoding. Decoder model complexity increases across rows while regularization strength β_{reg} increases moving down columns. (b) Latent space metrics as in Fig. 3 but for VAE models trained on independent 2D Gaussian data are plotted versus regularization strength. Error bars represent the standard deviation over six independent training runs.

for a VAE with an MSE decoder. While the natural coordinate system is clearly polar due to the spherical symmetry, and the encoders are capable of learning non-linear mappings to radial distances or angles, training consistently favors reconstructions of Cartesian coordinates based on a single latent coordinate that encodes some combination of Cartesian values. Specifying that the decoding probability distributions should apply to polar coordinates instead leads all decoder models but MSE to learn a latent space matching the radial direction (Fig. 6b). As the data are uniform in the angular coordinate, VAEs focus on modeling the bimodal radial probability density, which cannot be modeled without latent space information except in the case of the flow decoder. The MSE decoder cannot differentiate radial values due to a variance along this coordinate smaller than 1. Instead, the MSE decoder focuses on the higher variance angular coordinate since it is possible for it to partition this space. Overall, much higher acceptance rates are observed for VAEs predicting polar coordinates, as shown in Fig. 6c. While average $\mathrm{JD}_{\mathrm{encode}}$ values drop for all models around $\beta_{reg} = 2$ and the latent space is no longer utilized, the powerful flow decoder continues to maintain high f_{acc} values.

Molecular systems, even simple ones like alanine dipeptide, increase both the dimensionality and complexity of the underlying probability density of the training data. With an internal coordinate system (bond-angle-torsion, or BAT, coordinates), modeling of the data's probability density is significantly improved, as shown by much higher f_{acc} values in Fig. 7 compared to translationally and rotationally aligned Cartesian atomic coordinates. Average $\mathrm{JD}_{\mathrm{encode}}$ values drop off more quickly with β_{reg} when predicting BAT coordinates for all models but the MSE decoder (Fig. 7). This suggests that modeling the probability density of BAT coordinates is an easier task, not requiring significant latent space usage for either the autoregressive or flow decoders. Easier modeling of BAT representations compared to Cartesian coordinates likely stems from reduced correlations between degrees of freedom and simpler distributions that better match the chosen decoder distributions (e.g., Gaussian or von Mises distributions, as described in Section II). Since modeling Cartesian data (with all or only heavy atoms) is more difficult, the MSE decoder more quickly gives up on using the latent space whereas all other decoder models maintain a similar $\mathrm{JD}_{\mathrm{encode}}$ for all $\beta_{reg} \geq 1$. In contrast to all other systems and models, values of $\mathrm{JD}_{\mathrm{prior}}$ increase with β_{reg} for the autoregressive and flow decoders predicting alanine dipeptide Cartesian coordinates (with or without hydrogens). In these training runs, slight increases in encoder distribution overlap (lower average $\mathrm{JD}_{\mathrm{encode}}$) allows the KL loss to decrease with increased β_{reg} even as $P(\mathbf{z})$ becomes less like a standard normal distribution. This means that conformational space is more sharply partitioned by the latent space but that configurations within each partitioned region are less precisely encoded. These results reinforce the use of both the average $\mathrm{JD}_{\mathrm{encode}}$ and $\mathrm{JD}_{\mathrm{prior}}$ to understand latent space structure.

Fig. 8 provides information on the probabilistic independence and usage of latent dimensions. As described in Section II, the total correlation is lower if latent dimensions are

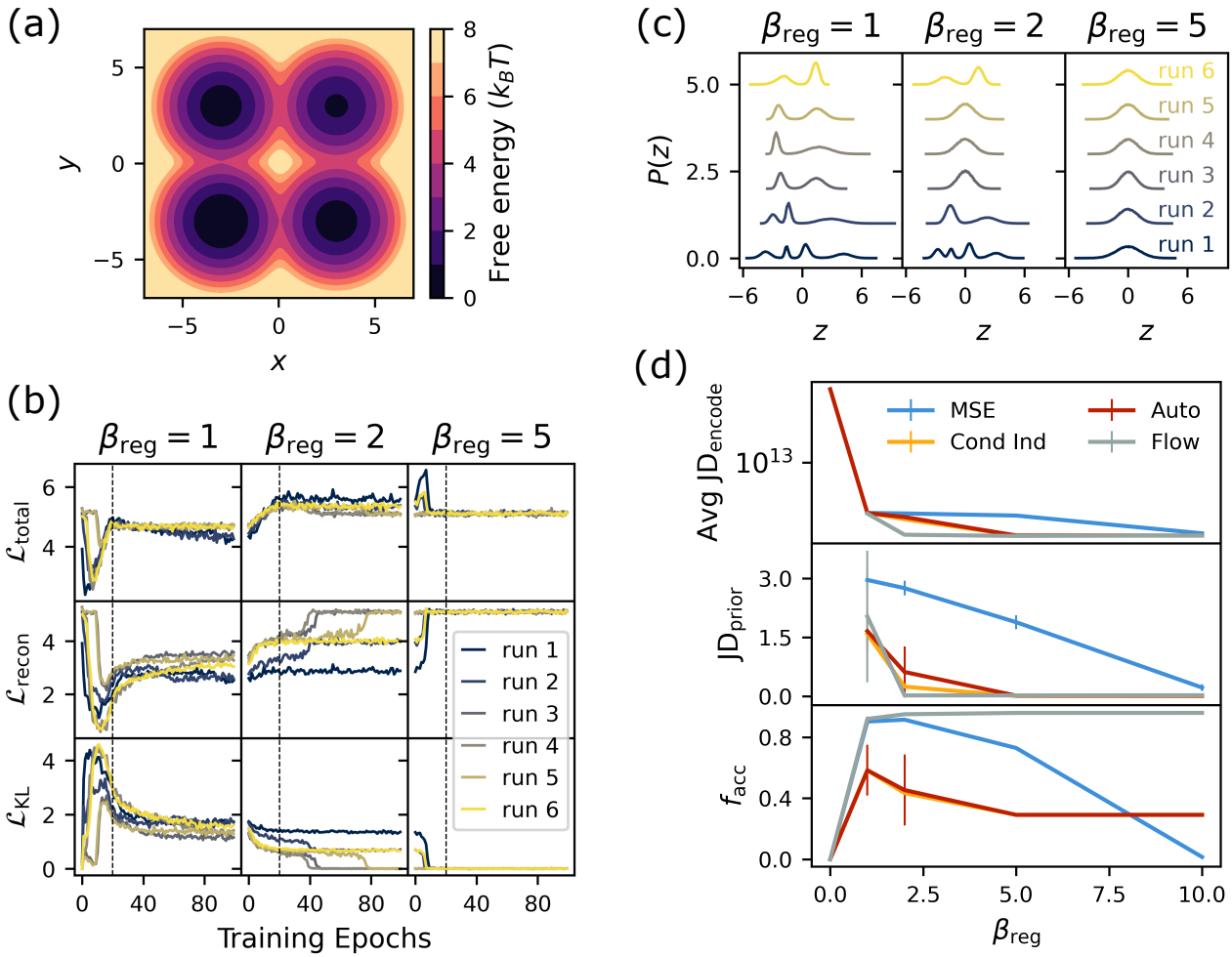

FIG. 5. (a) The free energy is shown for independent, identical Gaussian mixtures in each dimension, as described in Section II C. Data generated from this distribution are used to train VAE models with varying decoders and regularization strengths. (b) Contributions to the loss throughout independent training runs are shown for VAEs with autoregressive decoding models. Vertical dashed lines indicate the end of the annealing period in moving from a lower to a higher value of β_{reg}, with training within each run occurring sequentially from one regularization strength to the next. (c) Learned latent distributions $P(\mathbf{z})$ are shown for all runs of the autoregressive decoding model at the ends of training periods at different regularization strengths. (d) Various latent space metrics defined in Section II B and in Fig. 3 are plotted versus regularization strength. Error bars represent the standard deviation over six independent training runs.

independent from each other in a probabilistic sense. This does not necessarily imply that these directions are orthogonal in a spatial sense, but does indicate that they encode unique information. As we increase β_{reg} we observe a decrease in the total correlation (Fig. 8), which is typical and is the basis for the idea that increased weighting of the KL loss term leads VAEs to learn more "disentangled" representations[15]. However, total correlation may decrease when the KL loss decreases due to lower utilization of latent dimensions, which does not necessarily lead to meaningful disentanglements. We have noted throughout that the average JD$_{\text{encode}}$ tends to indicate if a latent space is utilized — if this quantity is low, encoding distributions overlap significantly and do not distinguish between input configurations. Examining data over all trained models for all systems (Fig. S5), a threshold of -1 for the base-10 logarithm of the average JD$_{\text{encode}}$ seems to distinguish between active and inactive latent spaces. We apply this threshold in a dimension-wise fashion to VAEs trained on alanine dipeptide data in the bottom panel of Fig. 8, calculating the average number of active latent dimensions versus β_{reg} for models with either 2 or 4 latents. With Cartesian coordinates, all models but the MSE decoder are able to maintain high numbers of active dimensions while total correlation decreases. The situation is reversed for BAT coordinates up to $\beta_{\text{reg}} = 5$, again indicating that this coordinate representation is easier to model and powerful decoders do not need to make as extensive use of the latent space. Techniques have also been proposed to enforce more meaningful disentanglements (i.e., independent latent dimensions) by only penalizing the total correlation and not the entire KL loss[18]. However, encoding independent noise is always a trivial solution to minimizing the total correlation, which may result from increasing the

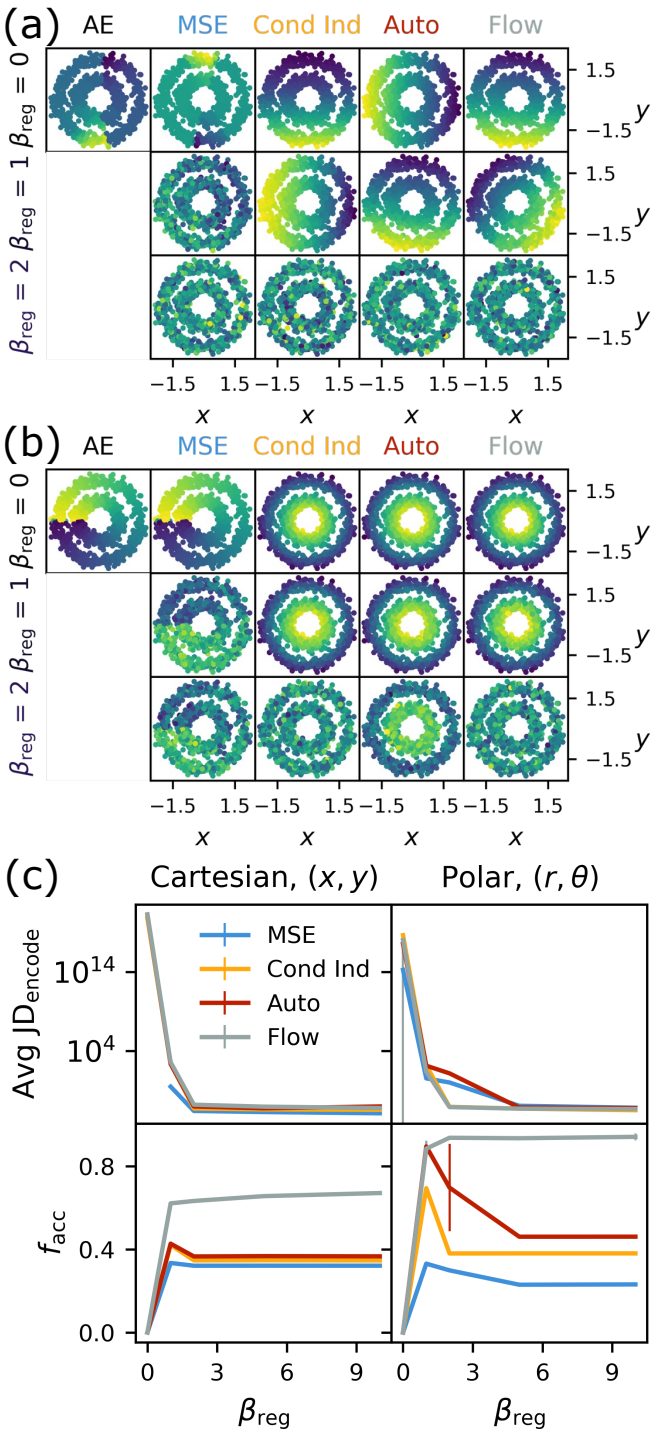

FIG. 6. Training data generated by a "Beltway" potential described in Section II C are colored by learned 1D latent values for VAEs trained to reproduce (a) Cartesian or (b) polar coordinates. AE indicates a standard autoencoder with no variational distribution (a deterministic encoding) and a mean-squared error loss for the decoding. Decoder model complexity increases across rows while regularization strength β_{reg} increases moving down columns. (c) Latent space metrics defined in Section II B and in Fig. 3 are plotted versus regularization strength for each coordinate system. Error bars represent the standard deviation over three independent training runs.

weight on both the full KL divergence or the total correlation alone. Further, as Locatello *et al.* [21] point out, even latent spaces with low total correlation may not be meaningfully disentangled to a human due to natural degeneracies, such as equivalent linear combinations. Indeed, 2D latent spaces learned for alanine dipeptide tend to distinguish free energy basins, but identify nonlinear combinations of the ϕ and ψ backbone dihedrals rather than cleanly separating them (Figs. S6-8). Error bars in Fig. 8 also indicate a large amount of variation over independent training runs, which was also observed by Locatello *et al.* [21] and led the authors to conclude that there is no significant difference in disentanglement between penalizing the full KL loss or only the total correlation.

IV. DISCUSSION

Across all systems, we observed many local minima due to significant compensation between reconstruction and KL divergence contributions to the loss. In some cases, these represent minima of indistinguishable total loss values, while in others the losses differ by small yet significant amounts, differentiating local from global minima. Either way, this presents difficulties with training VAEs and can impact the learned latent space, which in turn will change the performance of an enhanced sampling protocol or the interpretations drawn from an encoding. Fu *et al.* [48] suggested more sophisticated annealing protocols involving both increasing and decreasing β_{reg} to alleviate these issues. In this spirit, we trained VAE models for the Mueller potential in reverse, starting with the trained models at $\beta_{reg} = 10$ and annealing back to $\beta_{reg} = 1$ in the same manner as described in Section II. Figs. S9-S11 reveal that results are reversible for the MSE and conditionally independent decoders, but not for the autoregressive and flow decoders. Moving back to $\beta_{reg} = 1$, the autoregressive decoder does not regain the latent structuring observed in the lowest free energy basin, instead keeping the KL term low at the expense of a higher reconstruction loss (Figs. S10-S11). This represents a degenerate minima, as the total loss averaged across three independent training runs varies by less then 1% between the forward and reverse annealing protocols. Based on Welch's t-test, the total losses do not significantly differ, with a p-value of 0.886 (Table S1). Similarly, the flow decoder produces solutions of equal total loss with many different latent structures, including encoding no information in the latent space, also indicating degenerate minima (Table S1). While annealing in both directions may thus help in identifying degenerate solutions, it does not change the fact that such solutions seem common in sufficiently powerful decoding models.

Nonetheless, adjusting the regularization strength tips the balance between the reconstruction and KL loss terms and can adjust the resolution of the latent features that are learned. This was consistently observed across all systems through merging of distinct populations in latent space and overall less specific encodings (i.e., decreasing JD_{encode}). Transitions, however, may be quite sharp depending on the system and decoding model, with latent features dropped suddenly. Another

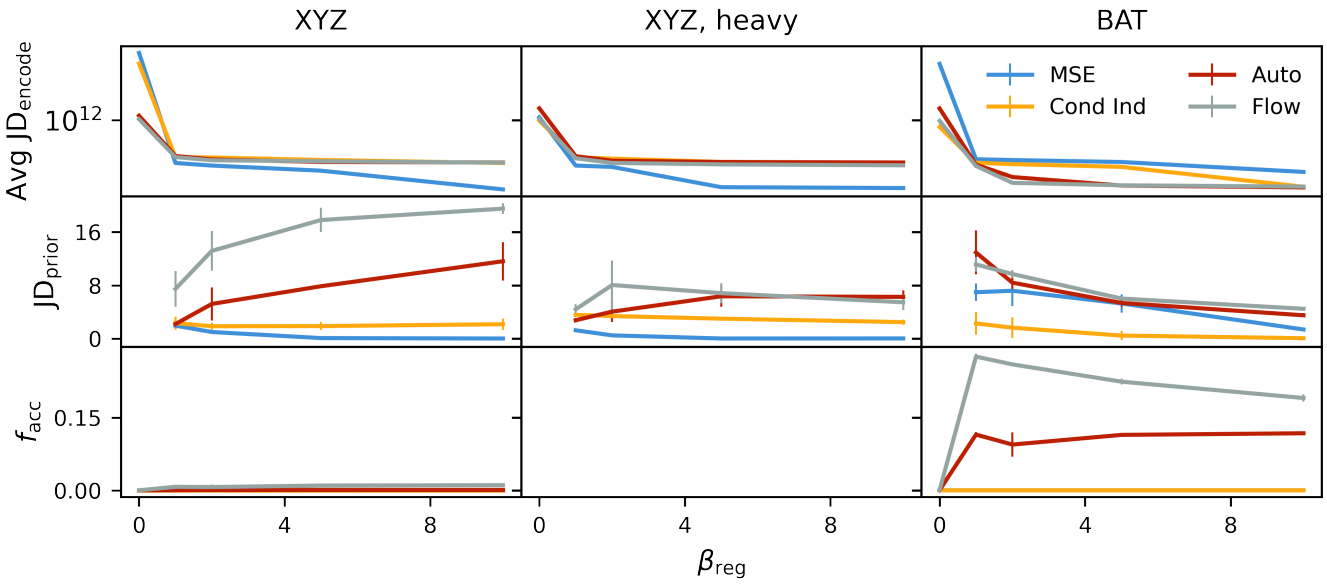

FIG. 7. Latent space metrics defined in Section II B and in Fig. 3 are plotted versus regularization strength for VAEs trained to model alanine dipeptide data in different coordinate systems. All results shown are based on VAE models with 2D latent spaces, though results are qualitatively similar in 1D and 4D cases (Fig. S4). VAE-based MC simulations are not possible for VAEs that only predict heavy-atom coordinates, and so f_{acc} values are not displayed for that case. Error bars represent the standard deviation over three independent training runs.

way to adjust learned latent features is of course through the choice of decoder model. While we have focused here on adjusting the modeling power of the decoding model, our results demonstrate that the focus should be on utilizing flexible probabilistic models with prior knowledge incorporated into restraints on their form. This is what led the less powerful MSE decoder to excellent performance on the toy Gaussian systems discussed in Figs. 4 and 5. A similar strategy was also outlined by Chen *et al.*[13] to tune the convolution filter size in autoregressive decoding models for images in order to focus the latent space on more global features of an image. In all of our autoregressive and flow decoders, we allow full coupling between degrees of freedom. We can easily modify this by adjusting masks on neural networks composing the autoregressive models so that only local correlations between spatially nearby degrees of freedom are built into the decoder. This represents a promising strategy for focusing VAEs on global CVs of biomolecules or self-assembling fluids.

Though we have employed highly flexible prior distributions throughout, more tailored models for $P(\mathbf{z})$ taking prior knowledge into account could also help focus attention on specific desirable aspects of CVs. For instance, applying Gaussian mixture priors and matching encoder distributions[10,11] explicitly encourages a VAE to learn latent spaces that cluster data. Priors with flows also allow for clustering, but this must be interrogated from the latent space after training and, as we saw in Fig. 5, learned clusters may be different with every training run due to many local minima in the optimization.

More flexible encoding distributions, such as those allowing multimodality, could also improve the expressivity of the latent encoding. This might be desirable in some cases, but generally the encoding should represent a compression and loss of information, and so simpler encoding distributions are desirable to focus on obtaining minimally complex yet maximally informative low-dimensional representations. Nonetheless, a relatively simple modification to enable simultaneous learning of both non-periodic and explicitly periodic latent coordinates, as discussed by Chen, Tan, and Ferguson[3] for standard autoencoders, would be to make the variational encoding distributions follow von Mises forms. This requires that sampling from the von Mises distribution be performed such that gradients of the sampling operation exist, which is possible through recently proposed, efficient, unbiased approximations of gradients[49]. The flows that we have applied can preserve periodicity[50], if present, further generalizing the prior in our VAE model. While relatively simple and worthy of study, the proposed switch from Gaussian encoding distributions would require investigations outside the scope of this work. Beyond these modifications, flows on priors could include full autoregression. We argue, however, that probabilistic independence of latent coordinates is a desirable quality and so encouraging this through conditionally independent encoding distributions and non-autoregressive flow is sufficient.

Most importantly, our results highlight the interplay between system complexity, decoder power, and latent space usage. We have shown that decoder models of intermediate modeling power, the conditionally independent and autoregressive models, tend to identify meaningful CVs over a wider range of β_{reg} values than either the less expressive MSE decoder or the powerful flow decoder. However, the specific regularization strength at which a model stops encoding information in the latent space depends on the specific system features that the decoder can capture. Across all systems and

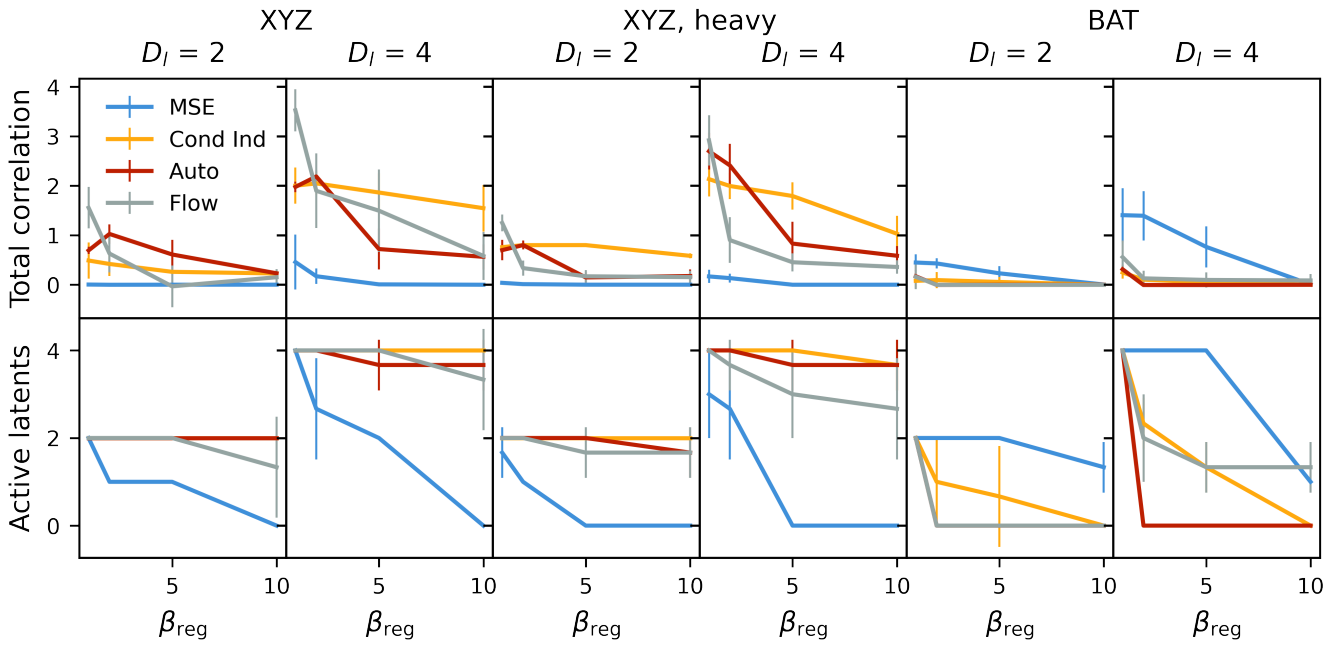

FIG. 8. (top) Total correlation assessing the probabilistic independence of latent space dimensions, and (bottom) estimates of the number of active latent dimensions based on the average number of latent dimensions with $\log_{10}$ of the average $\text{JD}_{\text{encode}}$ less than -1. For all VAEs trained on alanine dipeptide data in various coordinate systems, results are shown for models with either 2 or 4 latent dimensions. Error bars represent the standard deviation over three independent training runs.

decoder models, $\text{JD}_{\text{encode}}$ sharply distinguishes latent space usage (Fig. S5). However, each transition of $\text{JD}_{\text{encode}}$ occurs at a different level of regularization strength depending on the system and model. Both the total loss and f_{acc} indicate the appropriateness of different decoders for modeling a particular system, though only f_{acc} can be compared across systems. Low f_{acc}, however, does not necessarily mean that a model fails to utilize the latent space. Modeling the Cartesian coordinates of alanine dipeptide results in low f_{acc} for all decoders but typically involves high latent space usage, likely because the difficulty of the modeling task means large improvements in the reconstruction loss are possible at a small cost to the KL loss. Alternatively, high f_{acc} can occur in situations where the latent space is not utilized, as is commonly observed with the powerful flow decoder or when the system is exceedingly simple, as in the case of 2D independent Gaussians. While $\text{JD}_{\text{encode}}$ identifies a utilized latent space, we are not aware of general metrics agnostic to system and model that can be used to identify generic boundaries to define when this will occur. Though intriguing and potentially useful, we expect metrics measuring the difficulty of the modeling task and model suitability to be elusive. As for f_{acc} and the total loss, such metrics will tend to depend on the specifics of the system, the chosen number of latent dimensions, and the specific probabilistic form of the decoder.

V. CONCLUSIONS

Our results clearly illustrate that, when an appropriate balance is found between reconstruction and KL loss terms, VAE training focuses on learning the probability density of the input data. This leads to many desirable features of learned CVs, including smoothness and less of a tendency than standard autoencoders to "memorize" data, which can result in unnecessarily (and misleadingly) structured latent spaces (e.g., for simple 2D Gaussian data shown in Fig. 4). With a flexible, learned prior, here implemented through normalizing flows, VAEs can learn complicated CVs and their probability density $P(\mathbf{z})$ (or equivalently the free energy along the CV) without introduction of prior assumptions, such as the number of expected clusters in the data. Our results demonstrate, however, that the learned latent space will only reflect information useful to the decoding model. This leads to unused latents for either too simple or overly powerful decoding probability distributions, as was particularly clear when applying the MSE and flow decoders to Mueller potential data (Fig. 2). Adjusting the regularization strength (relative weight of reconstruction and KL loss terms) can help ensure that a VAE learns more than noise for a CV and can also help tune the resolution of learned low-dimensional features, or structure of the latent space. Enabling these insights, especially in more complicated molecular systems like alanine dipeptide, is the set of quantitative metrics we present here for characterizing VAE performance and latent space structure. In particular, $\text{JD}_{\text{encode}}$ usefully probes latent space usage and f_{acc} assesses

VAE model quality.

Decoder power itself can also tune the resolution of learned features within the latent space, which suggests promising opportunities in designing decoder probabilistic structures that incorporate knowledge of molecular physics to focus learning of CVs on global, long-range structure of the data. At a minimum, however, we recommend including prediction of decoder variances and caution against fixing the variance of the decoder model (i.e., using the MSE loss). Such assumptions are likely to lead to blurring of multimodal features in the latent space (e.g., Fig. 2b) and the resulting models will not be capable of capturing the common scenario of multiple free energy basins with differing shapes and depths, as was observed for both the Mueller and "Beltway" potentials. MSE decoders only outperformed other models for the cases of 2D independent Gaussians and Gaussian mixtures (Figs. 4-5), where the assumption in the MSE model of unit variance exactly matched the data. If there is good reason to believe a degree of freedom has a unimodal, approximately Gaussian distribution in a simulation, a fixed variance decoder with whitened input may be appropriate, though we argue this is relatively uncommon in simulations of molecular fluids or biomolecules. While normalizing flows represent a powerful method for modeling probability densities, we also encourage moderation in their use within decoders. In most systems studied here, flow decoders increase f_{acc} by a factor of at least 2 over even autoregressive decoders. However, they also frequently drive the average $\text{JD}_{\text{encode}}$ to less than 10^{-1}, indicating an unused latent space at even low values of β_{reg}, with such scenarios appearing as degenerate solutions with low KL loss contributions. Future research is necessary to identify how such powerful decoding models can be applied (e.g., designing a flow for local versus global probabilistic dependence) while ensuring that the latent space is utilized. Decoding models more tailored for molecular simulation applications may also alleviate difficulties associated with increasing numbers of degenerate minima with higher decoder complexity.

SUPPLEMENTARY MATERIAL

Supplementary figures and tables as well as schematics detailing the architectures of all models.

ACKNOWLEDGMENTS

This work was performed while J.I.M. held an NRC Postdoctoral Fellowship at the National Institute of Standards and Technology. The authors gratefully acknowledge computing time on the NIST Enki HPC cluster.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

[1] P. Gkeka, G. Stoltz, A. Barati Farimani, Z. Belkacemi, M. Ceriotti, J. D. Chodera, A. R. Dinner, A. L. Ferguson, J.-B. Maillet, H. Minoux, C. Peter, F. Pietrucci, A. Silveira, A. Tkatchenko, Z. Trstanova, R. Wiewiora, and T. Lelièvre, "Machine Learning Force Fields and Coarse-Grained Variables in Molecular Dynamics: Application to Materials and Biological Systems," Journal of Chemical Theory and Computation **16**, 4757–4775 (2020).

[2] A. Glielmo, B. E. Husic, A. Rodriguez, C. Clementi, F. Noé, and A. Laio, "Unsupervised Learning Methods for Molecular Simulation Data," Chemical Reviews **121**, 9722–9758 (2021).

[3] W. Chen, A. R. Tan, and A. L. Ferguson, "Collective variable discovery and enhanced sampling using autoencoders: Innovations in network architecture and error function design," The Journal of Chemical Physics **149**, 072312 (2018).

[4] J. M. L. Ribeiro, P. Bravo, Y. Wang, and P. Tiwary, "Reweighted autoencoded variational Bayes for enhanced sampling (RAVE)," The Journal of Chemical Physics **149**, 072301 (2018).

[5] T. Lemke and C. Peter, "EncoderMap: Dimensionality Reduction and Generation of Molecule Conformations," Journal of Chemical Theory and Computation **15**, 1209–1215 (2019).

[6] M. Chakraborty, C. Xu, and A. D. White, "Encoding and selecting coarse-grain mapping operators with hierarchical graphs," J. Chem. Phys. , 9 (2018).

[7] W. Wang and R. Gómez-Bombarelli, "Coarse-graining auto-encoders for molecular dynamics," npj Computational Materials **5**, 125 (2019).

[8] M. M. Sultan, H. K. Wayment-Steele, and V. S. Pande, "Transferable Neural Networks for Enhanced Sampling of Protein Dynamics," Journal of Chemical Theory and Computation **14**, 1887–1894 (2018).

[9] C. Wehmeyer and F. Noé, "Time-lagged autoencoders: Deep learning of slow collective variables for molecular kinetics," The Journal of Chemical Physics **148**, 241703 (2018).

[10] Y. B. Varolgünes, T. Bereau, and J. F. Rudzinski, "Interpretable embeddings from molecular simulations using Gaussian mixture variational autoencoders," Mach. Learn. , 21 (2020).

[11] M. Ghorbani, S. Prasad, J. B. Klauda, and B. R. Brooks, "Variational embedding of protein folding simulations using Gaussian mixture variational autoencoders," The Journal of Chemical Physics **155**, 194108 (2021).

[12] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," arXiv:1312.6114 [cs, stat] (2014), arXiv:1312.6114 [cs, stat].

[13] X. Chen, D. P. Kingma, T. Salimans, Y. Duan, P. Dhariwal, J. Schulman, I. Sutskever, and P. Abbeel, "Variational Lossy Autoencoder," arXiv:1611.02731 [cs, stat] (2017), arXiv:1611.02731 [cs, stat].

[14] S. Kullback and R. A. Leibler, "On Information and Sufficiency," The Annals of Mathematical Statistics **22**, 79–86 (1951).

[15] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "β-VAE: LEARNING BASIC VISUAL CONCEPTS WITH A CONSTRAINED VARIATIONAL FRAMEWORK," in *5th International Conference on Learning Representations* (Toulon, France, 2017).

[16] W. Chen, H. Sidky, and A. L. Ferguson, "Capabilities and limitations of time-lagged autoencoders for slow mode discovery in dynamical systems," The Journal of Chemical Physics **151**, 064123 (2019).

[17] C. X. Hernández, H. K. Wayment-Steele, M. M. Sultan, B. E. Husic, and V. S. Pande, "Variational Encoding of Complex Dynamics," Physical Review E **97**, 062412 (2018), arXiv:1711.08576.

[18] R. T. Q. Chen, X. Li, R. Grosse, and D. Duvenaud, "Isolating Sources of Disentanglement in Variational Autoencoders," arXiv:1802.04942 [cs, stat] (2019), arXiv:1802.04942 [cs, stat].

[19] H. Kim and A. Mnih, "Disentangling by Factorising," arXiv:1802.05983 [cs, stat] (2019), arXiv:1802.05983 [cs, stat].

[20] A. Kumar, P. Sattigeri, and A. Balakrishnan, "VARIATIONAL INFERENCE OF DISENTANGLED LATENT CONCEPTS FROM UNLABELED OBSERVATIONS," in *6th International Conference on Learning Representations* (Vancouver, BC, Canada, 2018).

[21] F. Locatello, S. Bauer, M. Lucic, G. Raetsch, S. Gelly, B. Schölkopf, and O. Bachem, "Challenging common assumptions in the unsupervised learning of disentangled representations," in *Proceedings of the 36th International Conference on Machine Learning*, Proceedings of Machine Learning Research, Vol. 97, edited by K. Chaudhuri and R. Salakhutdinov (PMLR, 2019) pp. 4114–4124.

[22]J. M. L. Ribeiro and P. Tiwary, "Toward Achieving Efficient and Accurate Ligand-Protein Unbinding with Deep Learning and Molecular Dynamics through RAVE," Journal of Chemical Theory and Computation **15**, 708–719 (2019).

[23]T. Lemke, A. Berg, A. Jain, and C. Peter, "EncoderMap(II): Visualizing Important Molecular Motions with Improved Generation of Protein Conformations," Journal of Chemical Information and Modeling **59**, 4550–4560 (2019).

[24]D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, "Improving Variational Inference with Inverse Autoregressive Flow," arXiv:1606.04934 [cs, stat] (2017), arXiv:1606.04934 [cs, stat].

[25]D. J. Rezende and S. Mohamed, "Variational Inference with Normalizing Flows," arXiv:1505.05770 [cs, stat] (2016), arXiv:1505.05770 [cs, stat].

[26]G. Papamakarios, E. Nalisnick, D. J. Rezende, S. Mohamed, and B. Lakshminarayanan, "Normalizing flows for probabilistic modeling and inference," Journal of Machine Learning Research **22**, 1–64 (2021), arXiv:1912.02762.

[27]M. D. Hoffman and M. J. Johnson, "ELBO surgery: Yet another way to carve up the variational evidence lower bound," in *30th Conference on Neural Information Processing Systems* (Barcelona, Spain, 2016) p. 4.

[28]C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios, "Neural Spline Flows," arXiv:1906.04032 [cs, stat] (2019), arXiv:1906.04032 [cs, stat].

[29]L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using Real NVP," arXiv:1605.08803 [cs, stat] (2017), arXiv:1605.08803 [cs, stat].

[30]J. V. Dillon, I. Langmore, D. Tran, E. Brevdo, S. Vasudevan, D. Moore, B. Patton, A. Alemi, M. Hoffman, and R. A. Saurous, "TensorFlow Distributions," arXiv:1711.10604 [cs, stat] (2017), arXiv:1711.10604 [cs, stat].

[31]G. Dorta, S. Vicente, L. Agapito, N. D. F. Campbell, and I. Simpson, "Structured Uncertainty Prediction Networks," arXiv:1802.07079 [stat] (2018), arXiv:1802.07079 [stat].

[32]G. Dorta, S. Vicente, L. Agapito, N. D. F. Campbell, and I. Simpson, "Training VAEs Under Structured Residuals," arXiv:1804.01050 [cs, stat] (2018), arXiv:1804.01050 [cs, stat].

[33]A. van den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu, "Conditional Image Generation with PixelCNN Decoders," arXiv:1606.05328 [cs] (2016), arXiv:1606.05328 [cs].

[34]M. Germain, K. Gregor, I. Murray, and H. Larochelle, "MADE: Masked Autoencoder for Distribution Estimation," arXiv:1502.03509 [cs, stat] (2015), arXiv:1502.03509 [cs, stat].

[35]A. B. Dieng, Y. Kim, A. M. Rush, and D. M. Blei, "Avoiding Latent Variable Collapse With Generative Skip Models," arXiv:1807.04863 [cs, stat] (2019), arXiv:1807.04863 [cs, stat].

[36]G. Papamakarios, T. Pavlakou, and I. Murray, "Masked Autoregressive Flow for Density Estimation," arXiv:1705.07057 [cs, stat] (2018), arXiv:1705.07057 [cs, stat].

[37]M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," (2015).

[38]D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," arXiv:1412.6980 [cs] (2017), arXiv:1412.6980 [cs].

[39]C. K. Sønderby, T. Raiko, L. Maaløe, S. K. Sønderby, and O. Winther, "Ladder Variational Autoencoders," arXiv:1602.02282 [cs, stat] (2016), arXiv:1602.02282 [cs, stat].

[40]H. Jeffreys, *Theory of Probability*, 3rd ed. (Clarendon Press, Oxford, 1961).

[41]J. I. Monroe and V. K. Shen, "Learning Efficient, Collective Monte Carlo Moves with Variational Autoencoders," Journal of Chemical Theory and Computation **18**, 3622–3636 (2022).

[42]F. Noé, S. Olsson, J. Köhler, and H. Wu, "Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning," Science **365**, eaaw1147 (2019).

[43]F. Noé, S. Olsson, J. Kohler, and W. Hao, "Boltzmann Generators – Sampling Equilibrium States of Many-Body Systems with Deep Learning," http://doi.org/10.5281/zenodo.3242635 (2021).

[44]P. Eastman, J. Swails, J. D. Chodera, R. T. McGibbon, Y. Zhao, K. A. Beauchamp, L.-P. Wang, A. C. Simmonett, M. P. Harrigan, C. D. Stern, R. P. Wiewiora, B. R. Brooks, and V. S. Pande, "OpenMM 7: Rapid development of high performance algorithms for molecular dynamics," PLOS Computational Biology **13**, e1005659 (2017).

[45]V. Hornak, R. Abel, A. Okur, B. Strockbine, A. Roitberg, and C. Simmerling, "Comparison of multiple Amber force fields and development of improved protein backbone parameters," Proteins: Structure, Function, and Bioinformatics **65**, 712–725 (2006).

[46]N. Michaud-Agrawal, E. J. Denning, T. B. Woolf, and O. Beckstein, "MDAnalysis: A toolkit for the analysis of molecular dynamics simulations," Journal of Computational Chemistry **32**, 2319–2327 (2011), https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcc.21787.

[47]R. J. Gowers, M. Linke, J. Barnoud, T. J. E. Reddy, M. N. Melo, S. L. Seyler, J. Domański, D. L. Dotson, S. Buchoux, I. M. Kenney, and O. Beckstein, "MDAnalysis: A Python Package for the Rapid Analysis of Molecular Dynamics Simulations," in *Proceedings of the 15th Python in Science Conference*, edited by S. Benthall and S. Rostrup (2016) pp. 98–105.

[48]H. Fu, C. Li, X. Liu, J. Gao, A. Celikyilmaz, and L. Carin, "Cyclical Annealing Schedule: A Simple Approach to Mitigating KL Vanishing," arXiv:1903.10145 [cs, stat] (2019), arXiv:1903.10145 [cs, stat].

[49]M. Figurnov, S. Mohamed, and A. Mnih, "Implicit Reparameterization Gradients," in *32nd Conference on Neural Information Processing Systems* (Montreal, Canada, 2018) p. 12.

[50]D. J. Rezende, G. Papamakarios, S. Racanière, M. S. Albergo, G. Kanwar, P. E. Shanahan, and K. Cranmer, "Normalizing Flows on Tori and Spheres," arXiv:2002.02428 [cs, stat] (2020), arXiv:2002.02428 [cs, stat].