Combined Representation and Generation with Diffusive State Predictive Information Bottleneck

Richard John
University of Maryland

Yunrui Qiu
University of Maryland

Lukas Herron
University of Maryland

Pratyush Tiwary*
University of Maryland

Abstract

Generative modeling becomes increasingly data-intensive in high-dimensional spaces. In molecular science, where data collection is expensive and important events are rare, compression to lower-dimensional manifolds is especially important for various downstream tasks, including generation. We combine a time-lagged information bottleneck designed to characterize molecular important representations and a diffusion model in one joint training objective. The resulting protocol, which we term Diffusive State Predictive Information Bottleneck (D-SPIB), enables the balancing of representation learning and generation aims in one flexible architecture. Additionally, the model is capable of combining temperature information from different molecular simulation trajectories to learn a coherent and useful internal representation of thermodynamics. We benchmark D-SPIB on multiple molecular tasks and showcase its potential for exploring physical conditions outside the training set.

1 Motivation

In various scientific areas, probability distributions of interest are inaccessible and characterized only by a set of known samples. Biomolecules like proteins, for example, are highly dynamic and can adopt diverse conformations that critically influence their biological functions. However, sampling the equilibrium distribution of protein structure is usually prohibitively costly, and even when simulation is feasible, it only yields the structure ensemble at a particular thermodynamic state (e.g., a certain temperature) [1–3]. Generating samples from the equilibrium distribution, is especially valuable when training data is scarce or difficult to obtain [4, 5]. This goal is addressed by a broad class of machine learning algorithms known as probabilistic generative models (PGMs). Motivated by widespread, successful application of PGMs to sampling problems, and the Information Bottleneck (IB) principle to learning physically-motivated representations, we introduce the Diffusive State Predictive Information Bottleneck (D-SPIB) and examine its performance on molecular modeling tasks. In addition, we describe how D-SPIB may infer the temperature dependence of metastable states from limited multi-temperature data. Temperature-aware generative models have been introduced in contemporary works, where they have been demonstrated to enhance sampling post-simulation [6–9].

Generative modeling in lower-dimensional spaces is advantageous. In machine learning generally, the 'curse of dimensionality' refers to a phenomenon where the amount of data required to train a model scales exponentially with data dimensionality. When the amount of training data is insufficient, the model overfits and exhibits undesirable behavior like memorization [10]. Indeed, theoretical study has identified a critical 'collapse time' in diffusion models–a popular class of PGMs–which indicates

*Email: ptiwary@umd.edu

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Frontiers in Probabilistic Inference.

memorization of the training set [11]. The phenomenon is corroborated by practical experiments, which have demonstrated that generative models become less accurate at density estimation as dimensionality increases when the amount of training data is held constant [12].

State Predictive Information Bottleneck (SPIB) provides a physically meaningful reduced-dimensional space. This physics-inspired framework of the variational IB finds a set of representations, i.e., collective variables, for molecular systems that capture the important slowest degrees of freedom and provide maximal information for dynamics propagation [13, 14]. The representations derived from SPIB have been validated on tasks such as molecular kinetics calculation [15] and exploration of aqueous crystal nucleation [16]. Recently, the Latent Thermodynamic Flows (LaTF) model was developed to simultaneously train a normalizing flow PGM and an SPIB model for unified generative modeling and representation learning [9].

Jointly learning to represent and generate data outperforms generation with pre-trained encodings. Variational autoencoders (VAEs) [17] aim to find a low-dimensional latent embedding from which the data may be reconstructed. As noted above, these latent spaces are strong candidates for generative modeling [18, 19]. Contemporary work indicates that simultaneous optimization of a generative model and latent variables outperforms serial optimization on reconstructive tasks [9, 20].

Based on the above criteria, we believe diffusion models and the SPIB latent space as implemented in the proposed D-SPIB framework constitute a suitable pairing for joint representation learning and generative modeling.

2 D-SPIB

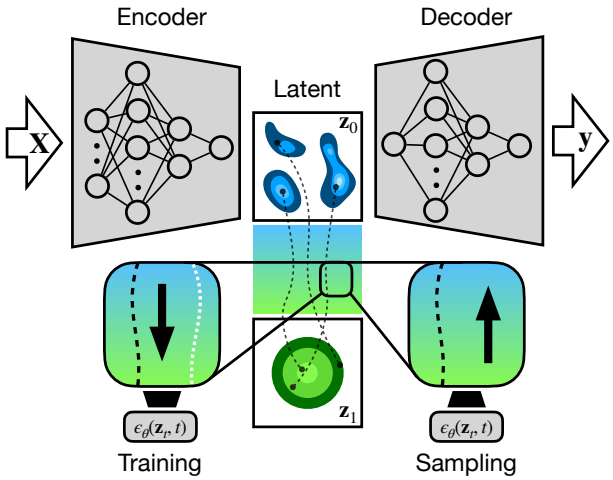

Figure 1: Diffusive SPIB architecture. Input data $\mathbf{X}$ is encoded to latent $\mathbf{z}_0$ and decoded to state label $\mathbf{y}$. The distribution of the encoded variable $\mathbf{z}_0$ is regularized by the IB prior distribution generated from $\mathbf{z}_1$ using a diffusion model with an easily sampled, pre-defined generative prior distribution. Blow-ups show the diffusion trajectories, including the reference forward trajectories used for training (white), the learned forward trajectories (black), and the backward trajectories employed for sampling (black).

The SPIB architecture consists of a time-lagged VAE trained to predict the state of a molecular system at future time $t + \tau$ based on high-dimensional molecular embeddings at time t. Increasingly expressive priors for the VAE are sought after as development of SPIB continues [13, 21]. Accordingly, we develop the D-SPIB model by incorporating a score-based generative model to generate a more flexible and expressive IB prior distribution that closely aligns with the SPIB-encoded distribution (see the architecture in Fig. 1).

The SPIB loss function is a variational approximation to the IB principle [22]:

$$\mathcal{L}_{\mathrm{SPIB}} = \frac{1}{N}\sum_{n=1}^{N}\int d\mathbf{z}\, p_\theta(\mathbf{z}|\mathbf{X}^n)\Big[-\log q_\theta(\mathbf{y}^{n+\tau}|\mathbf{z}) + \beta \log\frac{p_\theta(\mathbf{z}|\mathbf{X}^n)}{r_\theta(\mathbf{z})}\Big] \tag{1}$$

where $p_\theta(\mathbf{z}|\mathbf{X}^n)$ is an encoder that maps molecular embeddings $\mathbf{X}^n \in \mathbb{R}^D$ to latent variable $\mathbf{z} \in \mathbb{R}^d$ and $q_\theta(\mathbf{y}^{n+\tau}|\mathbf{z})$ is a decoder that predicts future-transition-state label $\mathbf{y}^{n+\tau} \in \mathbb{R}$ from $\mathbf{z}$. The lag time τ enables the model to learn representations relevant for long-term dynamics, while the regularization factor β constrains the encoded distribution $p_\theta(\mathbf{z}|\mathbf{X}^n)$ to remain close to the prior $r_\theta(\mathbf{z})$.

We present the D-SPIB objective function (derived in full in Appendix A):

$$\mathcal{L}_{\mathrm{D-SPIB}} = \frac{1}{N}\sum_{n=1}^{N}\Big[\mathbb{E}_{p_\theta(\mathbf{z}_0|\mathbf{X}^n)}\Big[-\log q_\theta(\mathbf{y}^{n+\tau}|\mathbf{z}_0) + \beta \log p_\theta(\mathbf{z}_0|\mathbf{X}^n)\Big] + \beta \cdot \mathbb{E}_{t\sim\mathcal{U}[0,1]}\Big[\frac{g(t)^2}{2}\mathbb{E}_{p_\theta(\mathbf{z}_t,\mathbf{z}_0|\mathbf{X}^n)}\Big[||\nabla_{\mathbf{z}_t}\log p(\mathbf{z}_t|\mathbf{z}_0) - \nabla_{\mathbf{z}_t}\log r_\theta(\mathbf{z}_t)||_2^2\Big]\Big]\Big] \tag{2}$$

In place of SPIB's standard prior, we employ a diffusion model to construct a more flexible and trainable prior distribution. Following Theorem 1 of [23], the regularization term in Eq. 1 can be written as a cross-entropy and further reformulated as a score-matching objective. Here, the latent variable $\mathbf{z}_t \equiv \mathbf{z}(t)$ is a time-dependent variable ($t \in [0, 1]$), where $\mathbf{z}_0$ denotes the SPIB-encoded variable and $\mathbf{z}_1$ the diffusion prior variable. The diffusion dynamics for $\mathbf{z}$ follow stochastic differential equations (SDEs):

$$\text{Forward: } d\mathbf{z} = f(\mathbf{z}, t)dt + g(t)d\mathbf{w} \tag{3}$$

$$\text{Reverse: } d\mathbf{z} = \big[f(\mathbf{z}, t) - g^2(t)\nabla_{\mathbf{z}}\log p_t(\mathbf{z}_t)\big]dt + g(t)d\bar{\mathbf{w}} \tag{4}$$

where $\mathbf{w}$ is the standard Wiener process. In the reverse equation, dt and the Wiener process $\bar{\mathbf{w}}$ are understood to run from $t = 1$ to $t = 0$. Coupled SDEs like the above are at the foundation of score-based models [24]. In broad strokes, the forward process transports samples from an unknown density $p_0(\mathbf{z}_0)$ to a Gaussian prior distribution $p_1(\mathbf{z}_1)$. The stochastic transport induces a series of marginal densities $p_t(\mathbf{z}_t)$ that determine the score, $\nabla_{\mathbf{z}}\log p_t(\mathbf{z}_t)$. Eq. 2 regresses $\nabla_{\mathbf{z}_t}\log r_\theta(\mathbf{z}_t)$ to the gradient of the marginal noising kernel $p(\mathbf{z}_t|\mathbf{z}_0)$, which is equivalent to the score and is entirely determined by the (pre-determined) functional forms of $f(t)$ and $g(t)$ [25]. Once the score $\nabla_{\mathbf{z}_t}\log r_\theta(\mathbf{z}_t)$ is learned, the reverse diffusion process (Eq. 4) can be simulated from randomly sampled $\mathbf{z}^* \sim p_1(\mathbf{z}_1)$ to generate samples in the latent space. Incorporating the diffusion model does not alter the IB principle, since $r_\theta(\mathbf{z})$ in Eq. 1 is simply modeled as $r_\theta(\mathbf{z}_0)$ in Eq. 2.

We further extend the representational capacity of D-SPIB by tempering the generative model, i.e., by adjusting the variance of the generative prior as a linear function of temperature [2, 3, 6, 26]. During training, the variance of the prior is scaled on a per-sample basis based on an associated temperature value. We also construct a temperature embedding and inject it into the denoising network via residual-like additions, thereby making the score explicitly temperature dependent (see more details in Appendix B and C). Tempering the generative prior in this way allows the D-SPIB model to learn the temperature dependence of the SPIB state populations and optimize the latent representation accordingly.

3 Results

3.1 Benchmarking D-SPIB on an Analytical Potential System

Our first experiment evaluates D-SPIB on a 'three-hole' potential in two dimensions [27]. A trajectory of a single particle was generated by simulating Langevin dynamics for 5×10^7 steps at temperature $T = 1/k_B$, and a randomly-selected fifth of the simulation trajectory segment was removed to form a validation set. A SPIB model was trained on the simulation data, then the D-SPIB model was optimized (see Appendix C). We compare the D-SPIB generated free energy landscape in latent space to the D-SPIB encoded validation set landscape in Fig. 2. D-SPIB recovers the three expected metastable states with visual agreement between the distributions. Table 1 quantifies the agreement via the symmetrized KL divergence between the encoded and D-SPIB generated distributions (with standard deviation over three runs shown); the divergence of a generic SPIB model is provided for reference, where generated data is probabilistically drawn from the vanilla SPIB analytical prior [13], without any generative model.

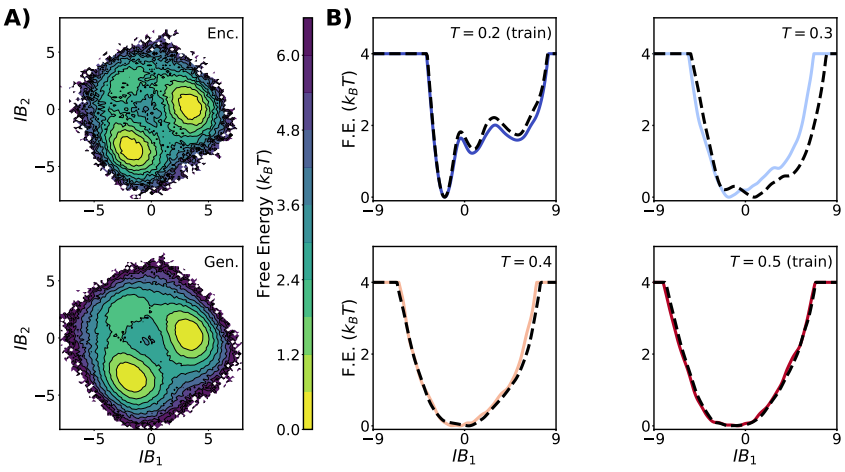

Figure 2: A) The distribution of encoded validation data and generated distribution by D-SPIB is shown for the three-hole potential. B) Free energy profiles along a D-SPIB latent dimension for generated data (colored, solid) and molecular dynamics data (black, dashed) in the multi-temperature LJ7 experiment.

Table 1: Single Temperature D_{KL}

System	D_{KL} (SPIB)	D_{KL} (D-SPIB)
Three-hole	9.773 ± 0.011	$\mathbf{0.042 \pm 0.002}$
LJ7 ($T = 0.2$)	15.76 ± 0.321	$\mathbf{0.760 \pm 0.123}$

3.2 Application of D-SPIB to Lennard-Jones Particle System

Our second experiment concerns a two-dimensional seven-particle system with interactions governed by the Lennard–Jones potential (LJ7 system)[28, 29]. Simulations were performed across temperatures $T \in \{0.2, 0.3, 0.4, 0.5\}$ (in units of ϵ/k_B). We first show numerical results comparing D-SPIB and generic SPIB for the single temperature case ($T = 0.2$) in Table 1 (with standard deviation over three independent runs reported), where we observe the better generation performance of the D-SPIB architecture on the symmetrized KL divergence test.

To examine model accuracy in generating data under new, unseen thermodynamic conditions, we train at pre-selected temperatures $T \in \{0.2, 0.5\}$ and compare to intermediate temperatures where no data was provided. In Fig. 2B, we compare the distribution generated by D-SPIB to a reference distribution obtained via molecular dynamics simulations across four temperatures: two that the model was trained on and two intermediate values, finding good agreement in each case. D-SPIB correctly predicts that the minima of the LJ7 system thermalize suddenly between $T = 0.2$ and 0.3 and then more gradually from $T = 0.3$ to 0.5 (see Fig. 2B).

4 Conclusion

In this work we have presented D-SPIB as a generative model with state-predictive capacity inherited from SPIB. We re-expressed the prior term in the SPIB objective function to derive the D-SPIB objective, and explained how the temperature dependence of the free energy may be inferred by learning temperature embedding and tempering the generative prior. D-SPIB's generative capabilities were evaluated on an analytic three-hole potential, where D-SPIB achieved a lower divergence between the encoded and generated latent distributions compared to SPIB. We further examined how well D-SPIB predicts the temperature dependence of the Boltzmann distribution of the LJ7 cluster of particles, and found that D-SPIB predicts rapid thermalization between temperatures of 0.2 and 0.3, which is supported by molecular dynamics simulation. Overall, we believe D-SPIB to be a suitable model for learning both informative representations and modeling the temperature dependence of systems in a highly data-efficient manner.

References

[1] Kresten Lindorff-Larsen, Stefano Piana, Ron O Dror, and David E Shaw. How fast-folding proteins fold. *Science*, 334(6055):517–520, 2011.

[2] Frank Noé, Simon Olsson, Jonas Köhler, and Hao Wu. Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457):eaaw1147, 2019.

[3] Manuel Dibak, Leon Klein, Andreas Krämer, and Frank Noé. Temperature steerable flows and boltzmann generators. *Physical Review Research*, 4(4):L042005, 2022.

[4] Pratyush Tiwary, Lukas Herron, Richard John, Suemin Lee, Disha Sanwal, and Ruiyu Wang. Generative artificial intelligence for computational chemistry: a roadmap to predicting emergent phenomena. *arXiv preprint arXiv:2409.03118*, 2024.

[5] Sarah Lewis, Tim Hempel, José Jiménez-Luna, Michael Gastegger, Yu Xie, Andrew YK Foong, Victor García Satorras, Osama Abdin, Bastiaan S Veeling, Iryna Zaporozhets, et al. Scalable emulation of protein equilibrium ensembles with generative deep learning. *Science*, page eadv9817, 2025.

[6] Lukas Herron, Kinjal Mondal, John S Schneekloth Jr, and Pratyush Tiwary. Inferring phase transitions and critical exponents from limited observations with thermodynamic maps. *Proceedings of the National Academy of Sciences*, 121(52):e2321971121, 2024.

[7] Suemin Lee, Ruiyu Wang, Lukas Herron, and Pratyush Tiwary. Exponentially tilted thermodynamic maps (exptm): Predicting phase transitions across temperature, pressure, and chemical potential. *arXiv preprint arXiv:2503.15080*, 2025.

[8] Eric R Beyerle and Pratyush Tiwary. Inferring the isotropic-nematic phase transition with generative machine learning. *Physical Review Letters*, 135(6):068102, 2025.

[9] Yunrui Qiu, Richard John, Lukas Herron, and Pratyush Tiwary. Latent thermodynamic flows: Unified representation learning and generative modeling of temperature-dependent behaviors from limited data. *arXiv preprint arXiv:2507.03174*, 2025.

[10] Alexandre B Tsybakov and Alexandre B Tsybakov. Nonparametric estimators. *Introduction to Nonparametric Estimation*, pages 1–76, 2009.

[11] Giulio Biroli, Tony Bonnaire, Valentin De Bortoli, and Marc Mézard. Dynamical regimes of diffusion models. *Nature Communications*, 15(1):9957, 2024.

[12] Richard John, Lukas Herron, and Pratyush Tiwary. A comparison of probabilistic generative frameworks for molecular simulations. *The Journal of Chemical Physics*, 162(11), 2025.

[13] Dedi Wang and Pratyush Tiwary. State predictive information bottleneck. *The Journal of Chemical Physics*, 154(13), 2021.

[14] Dedi Wang, Yunrui Qiu, Eric R Beyerle, Xuhui Huang, and Pratyush Tiwary. Information bottleneck approach for markov model construction. *Journal of chemical theory and computation*, 20(12):5352–5367, 2024.

[15] Suemin Lee, Dedi Wang, Markus A Seeliger, and Pratyush Tiwary. Calculating protein–ligand residence times through state predictive information bottleneck based enhanced sampling. *Journal of Chemical Theory and Computation*, 20(14):6341–6349, 2024.

[16] Ruiyu Wang, Shams Mehdi, Ziyue Zou, and Pratyush Tiwary. Is the local ion density sufficient to drive nacl nucleation from the melt and aqueous solution? *The Journal of Physical Chemistry B*, 128(4):1012–1021, 2024.

[17] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[18] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[19] Konpat Preechakul, Nattanat Chatthee, Suttisak Wizadwongsa, and Supasorn Suwajanakorn. Diffusion autoencoders: Toward a meaningful and decodable representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10619–10629, 2022.

[20] Guangyi Liu, Yu Wang, Zeyu Feng, Qiyu Wu, Liping Tang, Yuan Gao, Zhen Li, Shuguang Cui, Julian McAuley, Zichao Yang, et al. Unified generation, reconstruction, and representation: generalized diffusion with adaptive latent encoding-decoding. *arXiv preprint arXiv:2402.19009*, 2024.

[21] Jakub Tomczak and Max Welling. Vae with a vampprior. In *International conference on artificial intelligence and statistics*, pages 1214–1223. PMLR, 2018.

[22] Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

[23] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in neural information processing systems*, 34:11287–11302, 2021.

[24] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

[25] Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

[26] Yihang Wang, Lukas Herron, and Pratyush Tiwary. From data to noise to data for mixing physics across temperatures with generative artificial intelligence. *Proceedings of the National Academy of Sciences*, 119(32):e2203656119, 2022.

[27] Philipp Metzner, Christof Schütte, and Eric Vanden-Eijnden. Illustration of transition path theory on a collection of simple examples. *The Journal of chemical physics*, 125(8), 2006.

[28] Gareth A Tribello, Michele Ceriotti, and Michele Parrinello. A self-learning algorithm for biased molecular dynamics. *Proceedings of the National Academy of Sciences*, 107(41):17509–17514, 2010.

[29] Peter Schwerdtfeger and David J Wales. 100 years of the lennard-jones potential. *Journal of Chemical Theory and Computation*, 20(9):3379–3405, 2024.

A Loss function derivation

We derive the D-SPIB objective function beginning with the generic SPIB loss function and using Thm. 1 of [23] to express the prior in terms of a diffusion loss.

The SPIB loss function is as follows:

$$\mathcal{L}_{SPIB} = \frac{1}{N}\sum_{n=1}^{N}\int d\mathbf{z}\, p_\theta(\mathbf{z}|\mathbf{X}^n)\Big[-\log q_\theta(\mathbf{y}^{n+\tau}|\mathbf{z}) + \beta \log \frac{p_\theta(\mathbf{z}|\mathbf{X}^n)}{r_\theta(\mathbf{z})}\Big] \tag{5}$$

We begin by decomposing the term involving the prior into a cross entropy:

$$\begin{aligned}
&\mathcal{L}_{D-SPIB} \\
&= \frac{1}{N}\sum_{n=1}^{N}\mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{X}^n)}\Big[-\log q_\theta(\mathbf{y}^{n+\tau}|\mathbf{z}) + \beta \log p_\theta(\mathbf{z}|\mathbf{X}^n) - \beta \log r_\theta(\mathbf{z})\Big] &(6)\\
&= \frac{1}{N}\sum_{n=1}^{N}\Big[\mathbb{E}_{p_\theta(\mathbf{z}|\mathbf{X}^n)}\Big[-\log q_\theta(\mathbf{y}^{n+\tau}|\mathbf{z}) + \beta \log p_\theta(\mathbf{z}|\mathbf{X}^n)\Big] \\
&+ \beta \cdot \mathrm{H}(p_\theta(\mathbf{z}|\mathbf{X}^n)||r_\theta(\mathbf{z}))\Big] &(7)\\
&= \frac{1}{N}\sum_{n=1}^{N}\Big[\mathbb{E}_{p_\theta(\mathbf{z}_0|\mathbf{X}^n)}\Big[-\log q_\theta(\mathbf{y}^{n+\tau}|\mathbf{z}_0) + \beta \log p_\theta(\mathbf{z}_0|\mathbf{X}^n)\Big] \\
&+ \beta \cdot \mathbb{E}_{t\sim\mathcal{U}[0,1]}\Big[\frac{g(t)^2}{2}\mathbb{E}_{p_\theta(\mathbf{z}_t,\mathbf{z}_0|\mathbf{X}^n)}\Big[||\nabla_{\mathbf{z}_t}\log p(\mathbf{z}_t|\mathbf{z}_0) - \nabla_{\mathbf{z}_t}\log r_\theta(\mathbf{z}_t)||_2^2\Big]\Big] \\
&+ const.\Big] &(8)
\end{aligned}$$

Where in Eq. 6, we have separated the posterior and prior into two terms, and in Eq. 7 we have expressed the expectation of the log-prior over the posterior as a cross-entropy. In Eq. 8, we rewrite the cross-entropy as a score-matching loss term, following the same relabeling of $\mathbf{z} \to \mathbf{z}_0$ and assumption of a fixed noising kernel $p(\mathbf{z}_t|\mathbf{z}_0)$ derived from the pair of SDEs defining a noising process to a Gaussian prior explained in Sec. 2. We note the existence of constant terms dependent on the noise schedule relevant for calculating the exact cross-entropy. Since the terms are not optimized we ignore them, giving Eq. 2.

The expectation over the new joint distribution decomposes as follows: $p_\theta(\mathbf{z}_t, \mathbf{z}_0|\mathbf{X}^n) = p(\mathbf{z}_t|\mathbf{z}_0)p_\theta(\mathbf{z}_0|\mathbf{X}^n)$. Before going any further, we choose our $f(\mathbf{z}, t)$ and $g(t)$ such that the diffusion process corresponds to the variance-preserving SDE (VP-SDE) commonly used in Denoising Diffusion Probabilistic Models (DDPMs):

$$f(\mathbf{z}, t) = -\frac{1}{2}\beta_{noise}(t)\mathbf{z} \tag{9}$$

$$g(t) = \sqrt{\beta_{noise}(t)} \tag{10}$$

$$\Rightarrow d\mathbf{z} = -\frac{1}{2}\beta_{noise}(t)\mathbf{z}dt + \sqrt{\beta_{noise}(t)}d\mathbf{w} \tag{11}$$

where $\beta_{noise}(t)$ is an arithmetic progression from $\beta_{noise}(0) = 1e-4$ to $\beta_{noise}(1) = 0.2$; the values are chosen so that the final noised distribution closely approximates a Gaussian distribution. In practice, we discretize the SDE as a Markov chain, as done in DDPM. By the closure of Gaussian distributions and the reparameterization trick, we may explicitly draw noised samples at any arbitrary time in our forward process via:

$$\mathbf{z}_t = \sqrt{\bar{\alpha}_t}\mathbf{z}_0 + (1-\bar{\alpha}_t)\epsilon, \;\; \epsilon \sim \mathcal{N}(0, I) \tag{12}$$

where $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. For the VP-SDE, the score is directly related to the noise via:

$$\nabla_{\mathbf{z}_t} \log p(\mathbf{z}_t|\mathbf{z}_0) = -\frac{\epsilon}{\sigma_t} = -\frac{\epsilon}{\sqrt{1-\bar{\alpha}_t}} \tag{13}$$

We arrive at our final loss function by expressing the score-matching objective as a noise-prediction one (they are related by the noise schedule), and approximating the expectation over the posterior distribution as a sum over data. We use the 'unweighted' version of the denoising loss, in which $g(t)^2/\sigma_t^2 = 1$, i.e.,

$$\begin{aligned}
&\mathcal{L}_{D-SPIB} \\
&= \frac{1}{N}\sum_{n=1}^{N}\Big[\mathbb{E}_{p_\theta(\mathbf{z}_0|\mathbf{X}^n)}\Big[-\log q_\theta(\mathbf{y}^{n+\tau}|\mathbf{z}_0) + \beta \log p_\theta(\mathbf{z}_0|\mathbf{X}^n)\Big] \\
&\quad + \beta \cdot \mathbb{E}_{t\sim U[0,1]}\Big[\frac{1}{2}\frac{g(t)^2}{\sigma_t^2}\mathbb{E}_{\substack{p_\theta(\mathbf{z}_t,\mathbf{z}_0|\mathbf{X}^n)\\ \epsilon\sim\mathcal{N}(0,I)}}\Big[||\epsilon - \epsilon_\theta(\mathbf{z}_t,t)||_2^2\Big]\Big]\Big] \qquad (14) \\
&= \frac{1}{N}\sum_{n=1}^{N}\Big[\mathbb{E}_{p_\theta(\mathbf{z}_0|\mathbf{X}^n)}\Big[-\log q_\theta(\mathbf{y}^{n+\tau}|\mathbf{z}_0) + \beta \log p_\theta(\mathbf{z}_0|\mathbf{X}^n)\Big] \\
&\quad + \beta \cdot \mathbb{E}_{t\sim U[0,1]}\Big[\frac{1}{2}\mathbb{E}_{\substack{p_\theta(\mathbf{z}_t,\mathbf{z}_0|\mathbf{X}^n)\\ \epsilon\sim\mathcal{N}(0,I)}}\Big[||\epsilon - \epsilon_\theta(\mathbf{z}_t,t)||_2^2\Big]\Big]\Big] \qquad (15)
\end{aligned}$$

While this does not follow exactly from Eqs. 10 and 13, we empirically observed that this choice resulted in more stable training.

B Training and sampling

Training of D-SPIB follows the standard SPIB training procedure (i.e., a self-consistent and iterative training approach) outlined in [13], with the exception of the new loss function. Training begins with multiple iterations of SPIB training, so that the latent space converges to a small number of relevant metastable states before the D-SPIB loss function is introduced. Otherwise convergence is hampered since the diffusion model is learning a 'moving target'. The D-SPIB loss is entirely compatible with the vanilla SPIB training, during which short-lived states are merged or dropped when irrelevant, as only the decoder matters for this task.

For multi-temperature training, the variance of the Gaussian prior target is scaled by the temperature value in both training and sampling. Temperature information is also provided to the denoising network via learned embeddings. Samples are generated by simulating the VP-SDE (see Alg. 1).

C Experimental details

In all experiments, the noise prediction network is a 7-layer fully connected sequential network of `torch.nn.Linear` layers, with `torch.nn.ReLU` activation functions. Diffusion time and simulation temperature are both embedded into the network by first taking Fourier feature projections and appending these features to the representation at each layer. Model hyperparameters for the two experiments in this work are listed in Table 2.

The two-dimensional three-hole potential is defined as follows:

$$V(x,y) = 3e^{-x^2-(y-\frac{1}{3})^2} - 3e^{-x^2-(y-\frac{5}{3})^2} - 5e^{-(x-1)^2-y^2} - 5e^{-(x+1)^2-y^2} + 0.2x^4 + 0.2(y-\frac{1}{3})^4 \tag{16}$$

Molecular dynamics simulation of a single particle with mass $m = 1$ is performed using the Langevin middle integrator. The system temperature is controlled at $1/k_B$, and reflective periodic boundary conditions were applied. The simulation is run for 5×10^7 integration steps, with particle coordinates recorded every 50 steps.

Algorithm 1 D-SPIB Sampling

Require: Trained $\epsilon_\theta(\mathbf{z}_t, t)$, samples $\mathbf{z}_1 \sim p_1(\mathbf{z_1})$, noise schedule β_t, diffusion timesteps T_{diff}, optional temperature T

$\alpha_t = 1 - \beta_t$

$\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$

$t \leftarrow T_{diff}$

$\mathbf{z}_T \leftarrow \mathbf{z}_1$

if T exists **then**

$\sigma^2 = T^2$

else

$\sigma^2 = 1$

end if

while $t > 0$ **do**

$\gamma \sim \mathcal{N}(0, \sigma^2 I)$

$\mathbf{z}_{t-1} \leftarrow \frac{1}{\sqrt{\alpha_t}}(\mathbf{z}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(\mathbf{z}_t, t)) + \sqrt{\beta_t} \cdot \gamma$

end while

$\mathbf{z}_0 \leftarrow \frac{1}{\sqrt{\alpha_1}}(\mathbf{z}_1 - \frac{1-\alpha_1}{\sqrt{1-\bar{\alpha}_1}}\epsilon_\theta(\mathbf{z}_1, 1))$

return $\mathbf{z}_0$

Table 2: Training Hyperparameters

	LJ7	Three well
Hyperparameter	Value	Value
Lag Time	1	20
Latent Dim.	2	2
Encoder Type	Linear	Linear
Batch Size	512	512
Tolerance	0.001	0.001
Patience	5	5
Refinements	10	10
Diffusion Patience	150	50
Diffusion Refinements	0	0
Random Seed	42	42
Learning Rate	0.001	0.001
Information Bottleneck β	1×10^{-5}	1×10^{-5}
Diffusion Steps	100	100

The Lennard-Jones 7 system consists of seven particles interacting via the Lennard-Jones potential in two dimensions. Six simulations were performed with a Langevin thermostat at temperatures from $0.2\epsilon/k_B$ to $0.7\epsilon/k_B$ in increments of $0.1\epsilon/k_B$. Each simulation ran for 10^7 steps, with particle coordinates recorded every 100 steps. Additional simulation details for these two systems can be found in Ref [9].

D Code availability

The code containing the model, datasets, experiments, and analysis is available at the following GitHub link: `https://github.com/rickyjohnwhu/diff-spib`.

E Acknowledgments

This research was entirely supported by the US Department of Energy, Office of Science, Basic Energy Sciences, CPIMS Program, under Award No. DE-SC0021009. We thank UMD HPC's Zaratan and NSF ACCESS (project CHE180027P) for computational resources. P.T. is an investigator at the University of Maryland Institute for Health Computing, which is supported by funding from

Montgomery County, Maryland and The University of Maryland Strategic Partnership: MPowering the State, a formal collaboration between the University of Maryland, College Park, and the University of Maryland, Baltimore.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We claim to introduce a combined representation learning and generative model, we show both of these capabilities on molecular datasets.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We describe theoretical limitations in the appendix, specifically the mismatch between DDPM VPSDE and the unweighted objective used.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: We do not prove any original theoretical results, but show the full derivation of our loss function. Additionally, we do not apply theorems from existing work outside their scope.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide the full anonymized code for our new model, as well as the datasets used in the experiments. Hyperparameters (including random seeds) are provided.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The full annotated codebase of the model, as well as datasets, are provided anonymously.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Hyperparameters are reported in the appendix. Certain machine learning details are omitted from the text for brevity, but are viewable in the code.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Standard deviations over three independent runs are reported in the text.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: The model does not require significant compute resources or specialized systems to run.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We reviewed the Code of Ethics and believe our work to be fully in compliance.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: The work published concerns molecular science and has little scope to affect society broadly.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The work is purely scientific and we see no risk of misuse that would need mitigation justifying safeguards.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Cited works are credited in references, no commercial assets requiring copyright considerations are used.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release licensable assets beyond an anonymized codebase.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No human subjects involved.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No human subjects involved.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLMs were used for auxiliary purposes but do not form an original part of the model introduced nor its analysis.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.