Predicting Phase Transitions Across Temperature, Pressure, and Chemical Potential Using Exponentially Tilted Thermodynamic Maps

Suemin Lee#
Biophysics Program and Institute for Physical Science and Technology, University of Maryland, College Park 20742, USA and University of Maryland Institute for Health Computing, Bethesda, Maryland 20852, USA

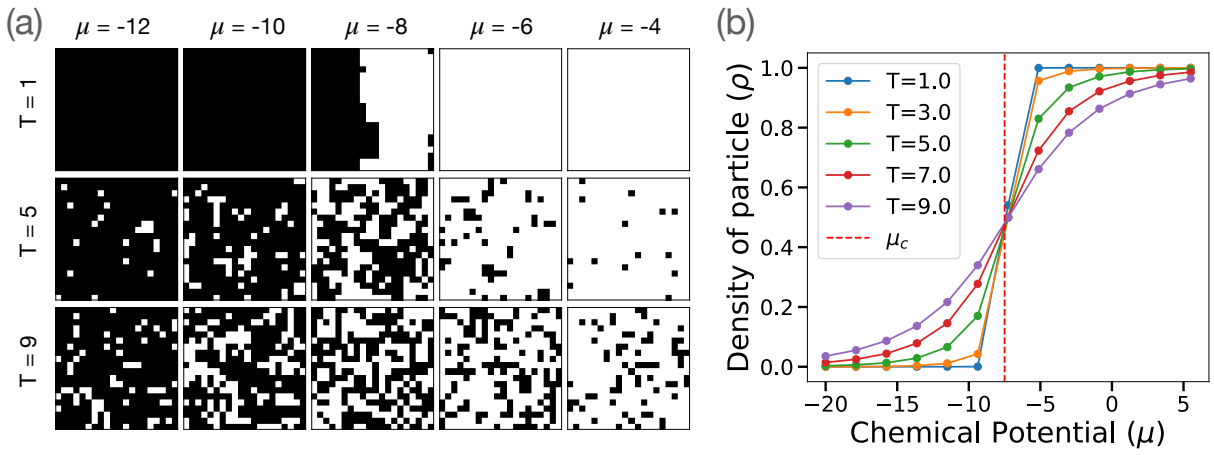
Ruiyu Wang#
Institute for Physical Science and Technology, University of Maryland, College Park 20742, USA

Lukas Herron
Biophysics Program and Institute for Physical Science and Technology, University of Maryland, College Park 20742, USA and University of Maryland Institute for Health Computing, Bethesda, Maryland 20852, USA

Pratyush Tiwary*
Biophysics Program and Institute for Physical Science and Technology, University of Maryland, College Park 20742, USA University of Maryland Institute for Health Computing, Bethesda, Maryland 20852, USA and Department of Chemistry and Biochemistry and Institute for Physical Science and Technology, University of Maryland, College Park 20742, USA

(Dated: August 25, 2026)

Supplementary Note 1. GRAND CANONICAL LATTICE GAS MODEL



Supplementary Fig. 1: Grand canonical Monte Carlo simulations of a 20×20 lattice gas at various pressures, chemical potentials, and temperatures: (a) representative configurations and (b) density change over averaged 10,000 samples.

The grand canonical lattice gas model described in the main text was studied using Monte Carlo (MC) simulations on a 20×20 square lattice. A total of 20,000 training samples were generated, comprising 10,000 configurations each at low and high temperatures. MC simulations ensure that configurations at a given temperature and chemical potential are sampled from the equilibrium distribution, accurately reflecting the conditions of the bath. The sampled output ranges from 0 to 1, representing the occupation probability of an atom at each lattice site.

Supplementary Fig. 1(a) illustrates the variations in lattice configurations under different pressures and chemical potentials, while Supplementary Fig. 1(b) shows the evolution of the system's density averaged over 5,000 samples across various temperatures and chemical potentials.

Supplementary Note 2. EXPONENTIALLY TILTED THERMODYNAMICS MAPS ALGORITHM

Here, we present a short algorithm of Exponentially tilted Thermodynamics Maps (expTM), which integrates thermodynamic control variables such as temperature (β) and another thermodynamic parameter of interest such as pressure or chemical potential (α) into a unified framework. The workflow begins by preparing input features with the associated thermodynamic conditions. Once it is prepared, the expTM model is then trained to map the data to an exponentially tilted Gaussian prior, where the tilting adjusts the prior distribution's mean and variance according to α and β, respectively. Finally, new samples are generated by reversing the diffusion process, allowing the model to generate new configurations under arbitrary thermodynamic conditions.

* ptiwary@umd.edu

Algorithm 1 Exponentially Tilted Thermodynamic Maps (expTM) with (α, β) Terms

1: **Step 1: Training data preparation.**

- Input feature configurations (or structures) $\{\mathbf{x}_i\}_{i=1}^N$.
- Associated thermodynamic bath parameters $\{\beta_i, \alpha_i\}_{i=1}^N$ (e.g., temperature β_i and pressure or chemical potential α_i).

Assign each configuration $\mathbf{x}_i$ its corresponding thermodynamic conditions (β_i, α_i) at which $\mathbf{x}_i$ was generated. Combine all configurations and construct training dataset: $\mathcal{D} = \big\{(\mathbf{x}_i, \beta_i, \alpha_i)\big\}_{i=1}^N$.

2: **Step 2: Estimate or initialize $\beta_i^{(j)}$ and $\alpha_i^{(j)}$.**

(a) **If** linear fluctuations are assumed:
$\beta_i^{(j)} \leftarrow \mathrm{Var}\big[\mathbf{x}_m^{(i)}\big]^{-1}, \quad \alpha_i^{(j)} \leftarrow \mathbb{E}\big[\mathbf{x}_m^{(i)}\big]$.

(b) **Else** keep them fixed at the input values:
$\beta_i^{(j)} = \beta, \quad \alpha_i^{(j)} = \alpha$.

3: **Step 3: Train the expTM model.**

(a) For a given thermodynamic condition (β, α), define a new conditional exponentially tilted Gaussian prior,
$p_{\text{prior}}(\mathbf{z} \mid \beta, \alpha) \propto \exp\Big(-\frac{\beta}{2}\big(\mathbf{z} - \alpha\big)^2\Big), \quad$ i.e., $\mathbf{z} \sim \mathcal{N}(\alpha, \beta^{-1})$.

This prior distribution is parameterized by (β, α) and is distinct from the bath parameters of $\{(\beta_i, \alpha_i)\}_{i=1}^N$ from the training data.

(b) Train a neural network and use score-matching to learn
$s_\theta(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \tau) \approx \nabla_{x,\alpha,\beta} \log p_\tau(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\beta})$, so the forward and reverse SDEs connect data $\mathbf{x}_m^{(i)}$ and prior distribution $p_{\text{prior}}(\mathbf{z} \mid \beta, \alpha)$.

(c) Minimize the training loss in the joint space $(\mathbf{x}, \alpha, \beta)$, so that the learned reverse SDE can be correctly mapped to the target data distributions at $p(\mathbf{x}, \beta, \alpha)$ sampled from the conditional tilted prior distribution $p(\mathbf{x} \mid \beta, \alpha)$.

4: **Step 4 : Generate New Samples.**

- Select thermodynamic parameter of interest (β^*, α^*). Sample $\widetilde{\mathbf{z}}$ from new tilted prior distribution $\widetilde{\mathbf{z}} \sim p_{\text{prior}}(\mathbf{z} \mid \beta^*, \alpha^*) = \mathcal{N}(\alpha^*, \beta^*)$.
- From the given joint state $(\widetilde{\mathbf{z}}, \beta^*, \alpha^*)$, simulate the reverse-time SDE towards the joint space $(\mathbf{x}, \beta, \alpha)$ using the learned score $s_\theta(\mathbf{x}, \beta, \alpha, \tau)$.

5: **Output:** A trained expTM model capable of generating samples at arbitrary thermodynamic conditions (T^*, α^*), including unseen values.

Supplementary Note 3. COMPUTATIONAL TIME COMPARISON BETWEEN MCMC AND EXPTM MODEL

To demonstrate the computational cost efficiency, we directly compared the computational time of MCMC sampling with the training and sampling time of our expTM model. For a fair comparison, all computations were carried out on a single NVIDIA A100 GPU throughout the test.

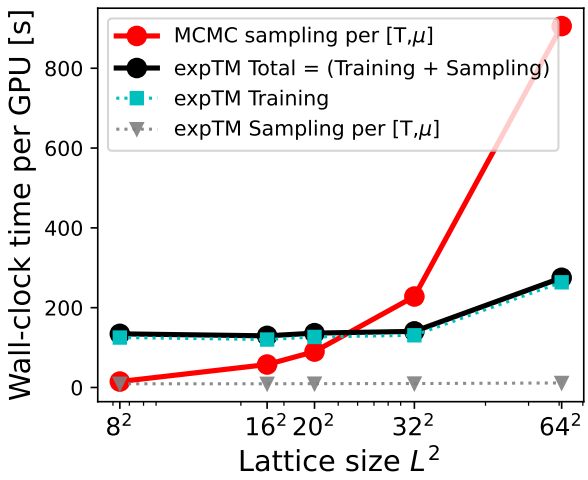
For MCMC, a single independent lattice configuration of a 20×20 system was generated by performing 250 equilibration steps followed by 200 production steps. For each [T,μ] pair, we generated 50 independent lattice configurations and computed the average density across them, which is the value shown in the heatmap. The total wall-clock time required to generate these 50 configurations and compute the averaged density for one [T,μ] point was approximately 79 seconds. Repeating this procedure for 625 distinct [T,μ] combinations resulted in a total wall-clock time of 49,597 seconds (approximately 13.7 hours).

In contrast, the diffusion model was trained within the SDE framework using an L_1-smooth loss for 10 epochs, which completed in approximately 183 seconds on the same GPU using data from only two training [T,μ] pairs. We used a VP-SDE (Variance preserving) discretization with $N_s = 100$ diffusion timesteps. Using this fixed discretization, we generated a single independent lattice configuration by running the reverse diffusion process for N_s steps. For each [T,μ] point, we generated 100 independent lattice configurations and computed the average density across them, which is the value reported for each [T,μ] point. The average sampling time required to generate these 100 configurations for one [T,μ] point was approximately 20 seconds, resulting in a total sampling time of 12,316 seconds for all 625 [T,μ] combinations. Since the training cost is negligible compared to the total sampling cost, the total computation time (training + sampling) is approximately 12,499 seconds (about 3.4 hours). A summary of the computational costs is provided in Supplementary Table I.

Method	**MCMC**	**expTM (tilted prior)**
Training time	N/A	183 s (10 epochs)
Sampling time per [T,μ] point	79 s	20 s
Total sampling time (625 points)	49,597 s	12,316 s
Total wall-clock time	49,597 s ($\sim$13.7 h)	12,499 s ($\sim$3.4 h)

Supplementary Table I: Wall-clock time comparison between MCMC sampling and expTM training and sampling, performed on a single NVIDIA A100 GPU.

To further demonstrate the computational efficiency, we compared performance across different lattice sizes, as shown in Supplementary Fig. 2. These results correspond to sampling a single [T,μ] point, where the reported values are averaged over 50 independent lattice configurations for MCMC and 100 configurations for the expTM model. The results show that the MCMC sampling time grows exponentially with lattice size L^2, whereas both the training and sampling times of the expTM model remain comparatively insensitive to system size until the lattice size reaches $L^2 = 64^2$. For small systems such as a $L^2 = 20^2$ size lattice at a single [T,μ] point, MCMC remains competitive. However, when sampling a large number of thermodynamic conditions, such as the 625 [T,μ] combinations considered



Supplementary Fig. 2: Wall-clock time per GPU required to generate samples using MCMC and diffusion-based (expTM) sampling as a function of lattice size L^2. For the diffusion approach, training time, sampling time, and total cost (training + sampling) are shown separately.

here, our expTM model provides a substantial computational advantage. Once trained, the expTM model enables repeated sampling at new thermodynamic conditions with no additional training cost, yielding a ~ 4 times reduction in wall-clock time at $L^2 = 20^2$, with an even greater advantage at larger lattice sizes.

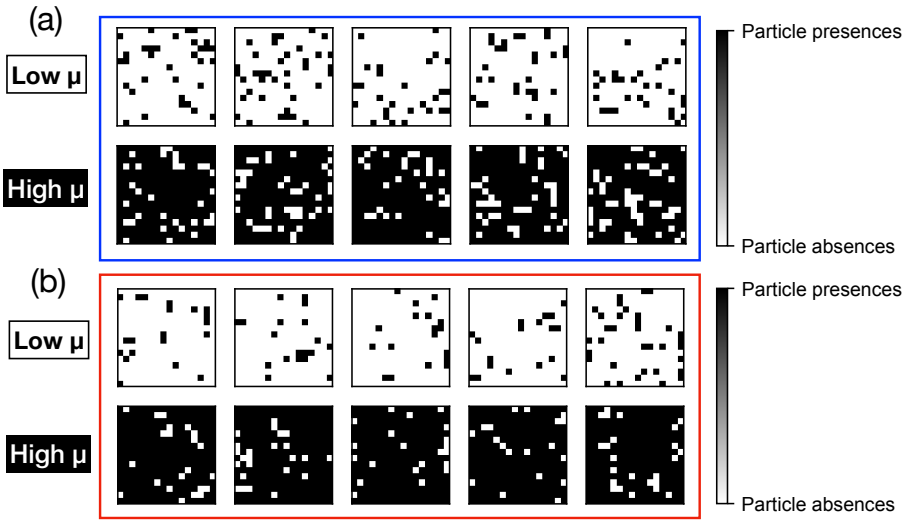
Supplementary Note 4. ABLATION EXPERIMENT ON LATTICE GAS MODEL

To demonstrate the effectiveness of our expTM model, we performed additional ablation experiments on the lattice gas system. First, we removed the conditioning on α and β, training the model to approximate the marginal distribution of the feature states without thermodynamic information. Second, we removed the exponentially tilted prior and instead mapped the data to a unit Gaussian distribution, thereby eliminating the thermodynamic parameterization in the prior distribution.

A. Removed α and β conditioning

One may question the importance of thermodynamic conditioning and how much the model deviates from a standard diffusion model. To examine this, we reverted to a simple diffusion model and trained it only on the lattice configurations x at two fixed thermodynamic conditions (i.e., chemical potentials μ), without embedding any thermodynamic variables. This model is analogous to a simple diffusion model, without labels. As a result, the model failed to generate samples, even for the training data. This was because the training data contained mixed information about the thermodynamic conditions. At that point, extrapolating or interpolating to different conditions became meaningless, as the model lost its ability to generate samples relevant to specific thermodynamic conditions. Thus, the test

clearly indicates that when the models are trained on only two fixed lattice configurations, they fail to produce meaningful results.



Supplementary Fig. 3: Ablation test on the lattice gas model. A standard conditional diffusion model was trained using configurations at two chemical potentials (low μ and high μ). (a) Representative lattice configurations from the training data at low and high chemical potentials. (b) Lattice gas configuration generated by the conditional diffusion model at the same two chemical potentials as in the training data.

To further validate this point, we tested a conditional diffusion model that behaves like a labeled diffusion model. In this case, the model embeds information regarding the chemical potential labels. As a result, the model can successfully recover the original training data at the two chemical potentials. However, it fails to generalize across thermodynamic conditions, particularly for unseen conditions during diffusion. The model cannot generate samples for unseen conditions because the corresponding labels do not exist. The results are shown in Supplementary Fig. 3, where the condition diffusion model can only retrieve the original training results.

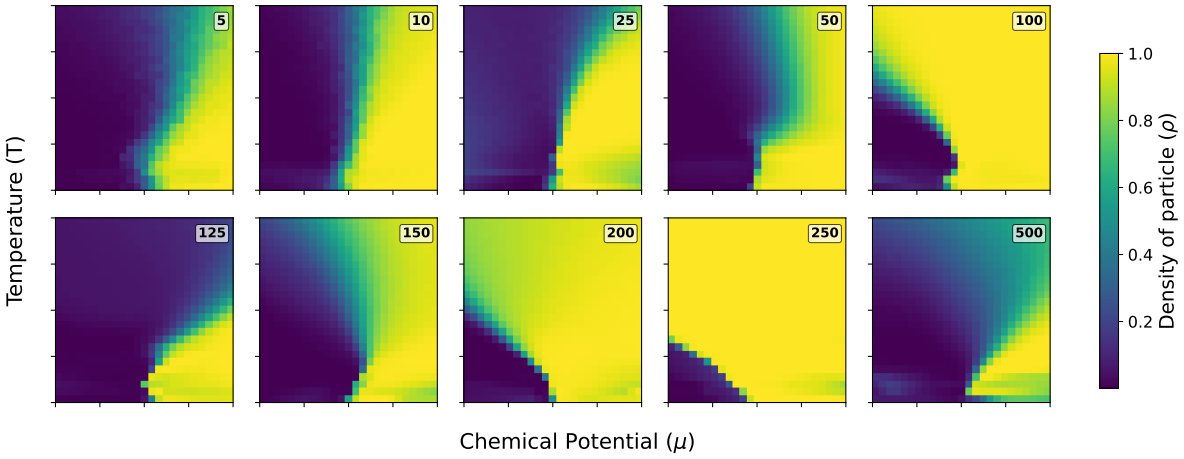
This highlights the strength of our expTM framework, which conditions not only on a single thermodynamic variable but on two variables simultaneously. The model is not implicitly learning temperature-dependent MCMC during the diffusion dynamics. Instead, the thermodynamic variables enter through the exponentially tilted prior and the joint score, allowing the model to correctly steer toward the target distribution and capture the nontrivial behavior of phase transitions.

B. Removed exponential tilting prior

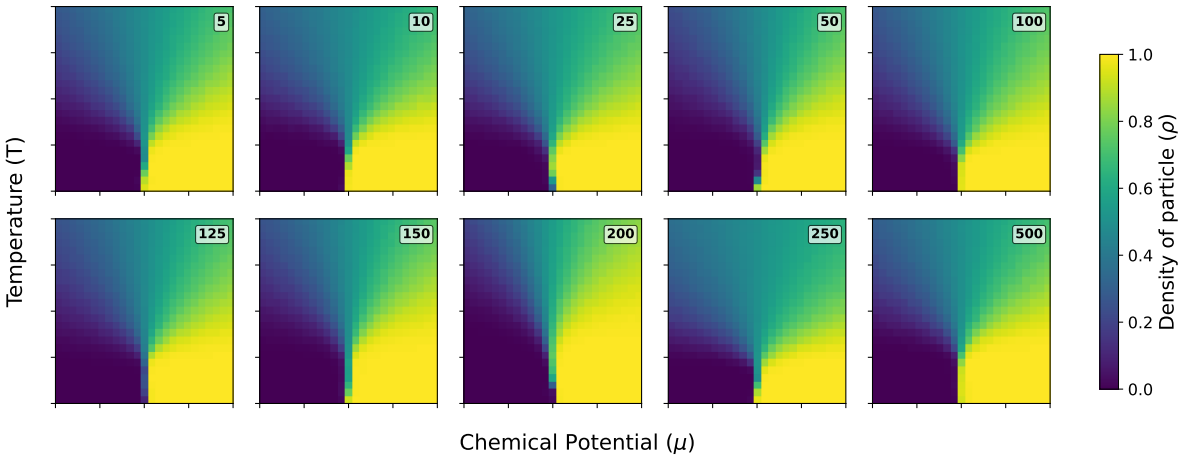
By construction, another key contribution of this work is the integration of an exponentially tilted prior by replacing the original unit normal Gaussian prior. To evaluate its role, we first revert the prior distribution to a unit normal Gaussian and examine the resulting generated samples in the absence of exponential tilting. In this setting, all training data are mapped to a single standard normal distribution, and the prior is no longer conditioned

on temperature or chemical potential. Consequently, the model effectively learns an unconditional distribution over the feature space x (or lattice configurations in the lattice gas model), rather than the conditional distribution $P(x \mid T, \mu)$. To ensure that the observed behavior is not due to insufficient training, we trained models over a wide range of epochs, e.g., epochs = [5, 10, 25, 50, 100, 125, 150, 200, 250, 500].

As shown in Supplementary Fig. 6, models trained with a unit normal prior fail to reproduce accurate phase diagrams, even at extended training times. In contrast, as shown in Supplementary Fig. 5, the exponentially tilted prior consistently yields accurate phase behaviors across all training epochs and random seeds.



Supplementary Fig. 4: Lattice generations using a unit normal Gaussian prior distribution under different numbers of epochs.



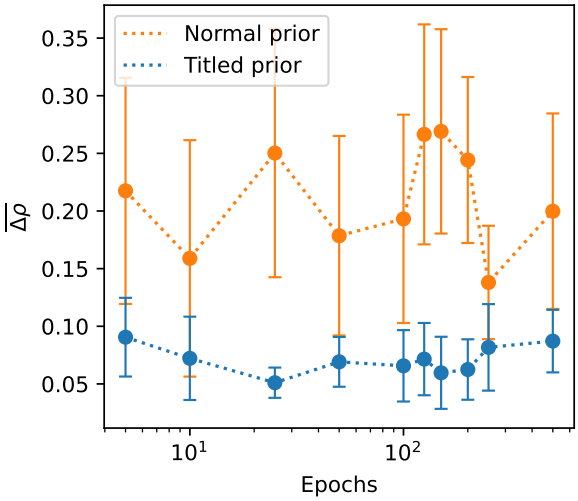
Supplementary Fig. 5: Lattice generations using exponentially tilted prior distribution under different numbers of epochs.

To further validate our results, we quantified the density error between the ground-truth MCMC results and samples generated by expTM using either a unit normal prior or the exponentially tilted prior, measured by $\Delta\rho$. For each case, lattice configurations on a 25×25

grid (625 points) were generated, and each model was retrained 10 times across all training epochs. We then computed the mean density error $\overline{\Delta\rho}$ and its variance.

Overall, the results indicate that the exponentially tilted prior achieves the lowest error at approximately 25 training epochs. In contrast, models trained with a unit normal prior exhibit larger mean errors and substantially higher variance across random seeds, demonstrating a lack of robustness and reproducibility relative to the ground-truth MCMC results.

This ablation study demonstrates the necessity of exponential tilting of the prior in the diffusion model. Without the tilted prior, even when conditioning parameters are applied to a unit Gaussian distribution—as in the original diffusion—the model fails to reproduce the correct phase transition behavior. The exponentially tilted prior is therefore a key component that enables correct conditional generation, allowing the model to represent thermodynamic conditions.



Supplementary Fig. 6: Mean density error $\overline{\Delta\rho}$ between generated samples and MCMC ground truth results as a function of training epochs, comparing models trained with a unit normal prior (orange) and an exponentially tilted prior (blue). Error bars indicate the standard deviation over 10 independent training runs.

For better comparison, we further computed the errors between the ground truth from the MCMC results, which is illustrated in Supplementary Fig. 6. Here, the error bars show the standard deviation across 10 training runs at each epoch.

Supplementary Note 5. ADDITIONAL SETTINGS AND RESULTS OF PHASE TRANSITION OF CO_2

A. Molecular dynamics simulations settings

The settings of molecular dynamics (MD) simulations for CO_2 are based on prior work.[1] All simulations are performed using GROMACS 2022.3,[2, 3] with each simulation containing 256 CO_2 molecules. The force field parameters for CO_2 are described in Supplementary Table II. To address simulation instabilities arising from the linear geometry of CO_2, two dummy atoms are added at a distance of $d_{CM} = 0.098925\,\text{nm}$ and with a mass of 22 using

the *virtual_sites2* keyword in GROMACS. The distance between the dummy atoms, $M1$ and $M2$, is constrained to 0.19785 nm using the *constraint* keyword in GROMACS. Simulations are conducted in the isothermal–isobaric (NPT) ensemble. The temperature is maintained at 350 K using the velocity-rescaling thermostat method with a relaxation time of 0.1 ps.[4] The pressures are set to 1, 3, 5, and 8 GPa across four simulations controlled by the Berendsen coupling algorithm.[5] The compressibility is set to 10^{-5} bar^{-1}. Each simulation runs for 500 ns with a time step of 2 fs. All bonds within the CO_2 molecules are constrained using the LINCS algorithm.[6]

Supplementary Table II: Force field parameters for CO_2 molecules.

σ_C (nm)	σ_O (nm)	ϵ_C (kJ/mol)	ϵ_O (kJ/mol)	q_C (e)	q_O (e)	d_{CO} (nm)
0.280	0.305	0.224	0.657	0.70	-0.35	1.160

MD simulations with enhanced sampling are carried out using the PLUMED package (version 2.8.1) patched into the MD engine.[7] The well-tempered metadynamics approach is used,[8] where a bias potential of 10 kJ/mol was deposited every 250 fs with a width of $\frac{2}{256}$. The virtual temperature in metadynamics is set to 450 K, with a bias factor of 200. To prevent extreme deformation of the simulation box, additional restraints were introduced. Specifically, a penalty potential (U_p) was imposed whenever any of the box dimensions (X, Y, or Z) exceeded 5.0 nm or fell below 1.7 nm:

$$U_{p_l} = \begin{cases} 1500 \times (l - 1.7)^2, & \text{if l<1.7} \\ 1500 \times (l - 5.0)^2, & \text{if l>5.0} \\ 0, & \text{if 1.7<l<5.0} \end{cases} \tag{S1}$$

where l represents the box length of the X, Y, or Z direction. An additional constraint on the total volume (V) was defined as

$$U_{pV} = 1500 \times (V - 11.5)^2, \quad \text{if } V > 11.5\,\text{nm}^3. \tag{S2}$$

Intramolecular orientation was chosen as collective variables (CVs), which captures the relative alignment between a given molecule and its neighbors. The CV compares these orientations to predefined target values in order to identify. [1, 9]

Two such CVs, λ_1 and λ_3, measure the average similarity of each CO_2 molecule to Phases I and III, respectively. The CV is defined as

$$\lambda = \sum_i^N \Gamma_i / N, \tag{S3}$$

where $N = 256$ and Γ_i is the similarity of molecule i:

$$\Gamma_i = \frac{\rho_i}{n_i} \sum_j^N f_{ij} \Theta_{ij}, \tag{S4}$$

with ρ_i and n_i determined by coordination number:

$$\rho_i = 1/[1 + e^{-b(n_i - n_{cut})}], n_i = \sum_{j \neq i} f_{ij} \tag{S5}$$

The term f_{ij} represent the continuous coordination weights,

$$f_{ij} = \frac{1}{1 + e^{a(r_{cut} - r_{ij})}}, \tag{S6}$$

where r_{ij} is the distance between molecule i and j, and $r_{cut} = 0.8$ nm. While Θ_{ij} quantifies how closely the relative orientation of neighboring molecules matches a reference phase:

$$\Theta_{ij} = \sum_{k}^{k_{max}} e^{-\frac{(\theta_{ij} - \theta_k)^2}{2\sigma_k^2}}, \tag{S7}$$

where $k_{max} = 2$ because multiple angles can describe the same phase. The parameter Γ_i approaches 1 if a molecule i perfectly aligns with a particular solid phase, while $\Gamma_i = 0$ indicates no alignment. The relevant reference angles of θ_k and σ_k are listed in Supplementary Table III.

Supplementary Table III: Values of θ_k and σ_k to calculate CVs λ_1 and λ_3.

	θ_1 (rad)	θ_2 (rad)	$\sigma_1 = \sigma_2$ (rad)	n_{cut}	r_{cut} (nm)
λ_1	1.23	1.90	0.25	5	0.4
λ_3	0.14	3.00	0.20	5	0.4

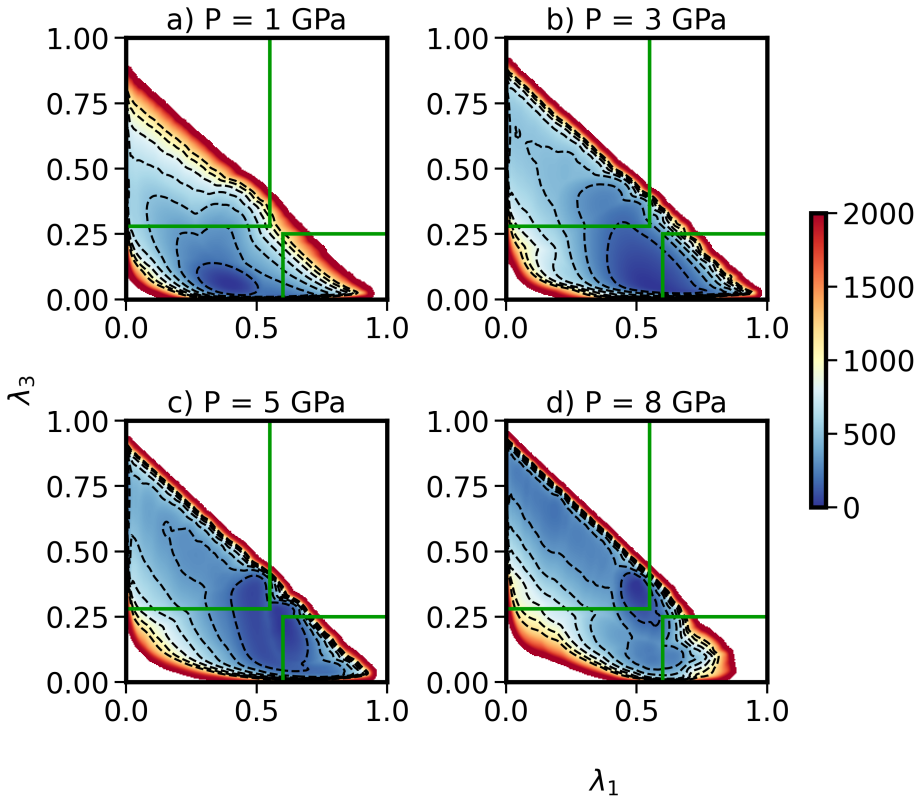
The free energy surface (FES), $F(\lambda_1, \lambda_3)$, was obtained using the PLUMED tool *sum_hills*. To determine the free energy difference from Phase I to Phase III (ΔG_{I_III}), we first defined the phase boundaries (Table I), as illustrated in the FES (Supplementary Fig. 7). CV values (λ_1, λ_3) outside the I/III regions were assigned to a liquid or amorphous phase. Finally, ΔG_{I_III} was calculated using:

$$\Delta G_{I\_III} = -RT \ln \frac{P(III)}{P(I)} = -RT \ln \frac{\iint_{(\lambda_1,\lambda_3)\in III} d\lambda_1 d\lambda_3 e^{-\frac{F(\lambda_1,\lambda_3)}{RT}}}{\iint_{(\lambda_1,\lambda_3)\in I} d\lambda_1 d\lambda_3 e^{-\frac{F(\lambda_1,\lambda_3)}{RT}}} \tag{S8}$$

B. Data preparation for the input of expTM

While phase similarity scores λ_1 and λ_3 serve as global descriptors to distinguish CO_2 Phases I and III, capturing molecular-level structural details is essential for providing expTM with an Ising-model-like representation. To achieve this, we use the molecule-wise order parameter Γ (Eq. S4). The input data for expTM are stored as NumPy arrays with shape $(n, 2, 16, 16)$, where n represents the number of selected configurations, the second dimension (2) corresponds to the Γ values for Phases I and III, and the (16×16) grid maps the 256 molecules in each configuration. For each pressure condition, we select 10^5 distinct configurations, weighting their occurrences according to the probability

$$P(\lambda_1, \lambda_3) = e^{-\frac{F(\lambda_1,\lambda_3)}{RT}} \tag{S9}$$

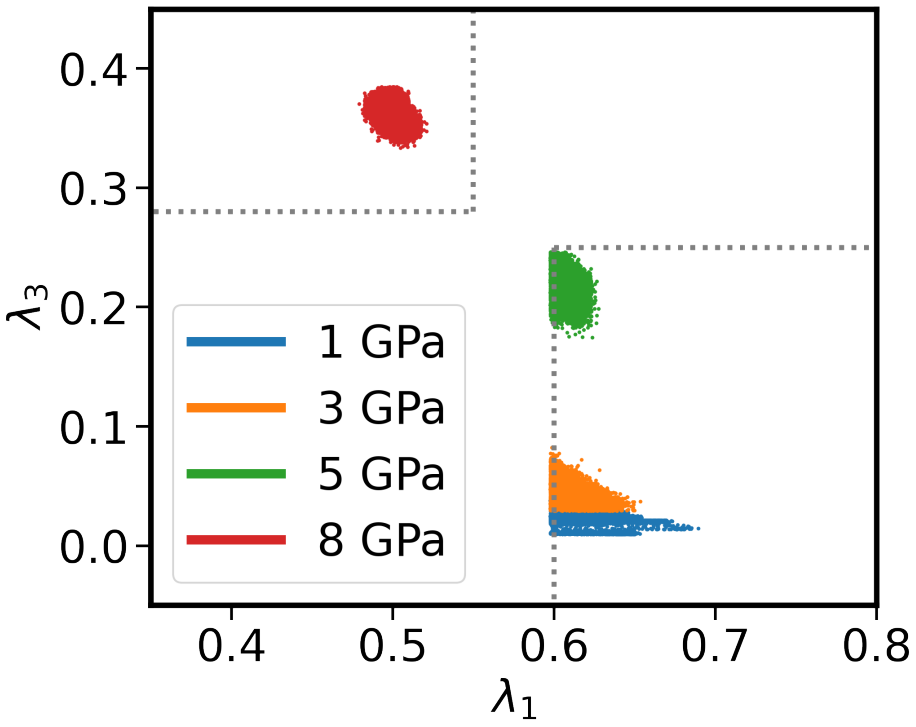


Supplementary Fig. 7: Free energy surfaces (FES) calculated from MD simulations of CO_2 under different pressures. The minimum value of the free energy is set to 0. The unit for the color bar is kJ/mol. The CVs on the X and Y axes are λ_1 and λ_3, respectively. The color transitions to white where the free energy $F(\lambda_1, \lambda_3) > 2000$ kJ/mol. The green line indicates the regions corresponding to different phases, which remain consistent across simulations at all pressures, as described in the text. The top-left corner represents Phase III, while the bottom-right corner corresponds to Phase I.

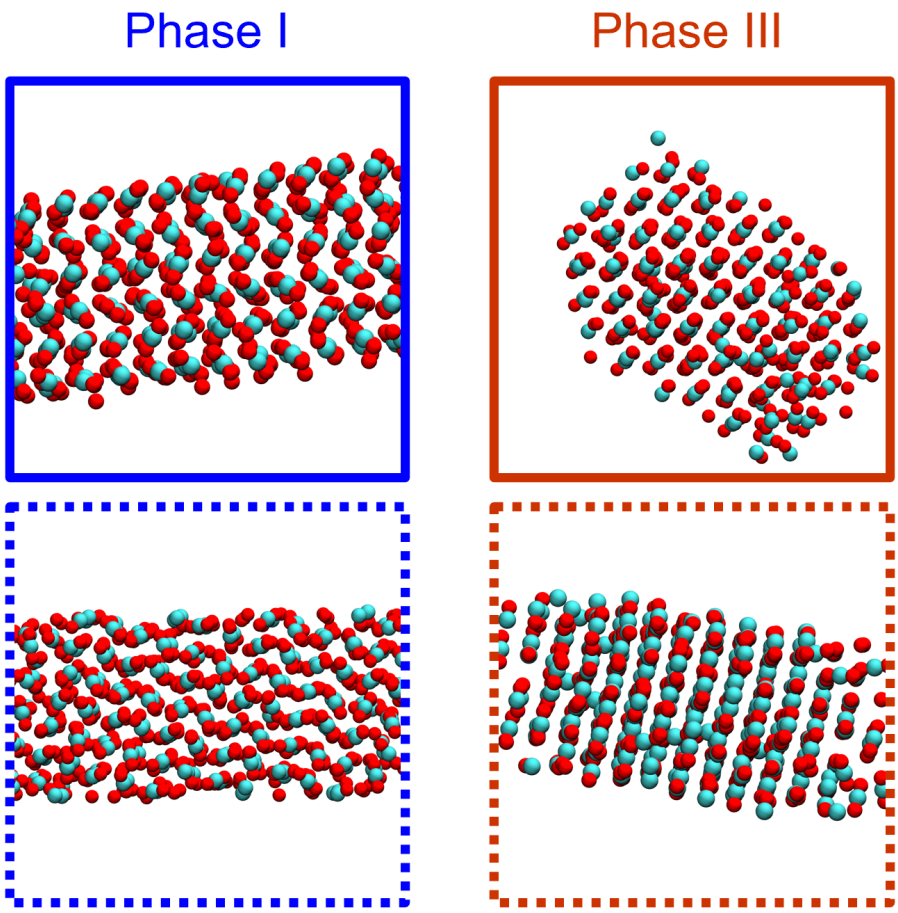
as illustrated in Supplementary Fig. 8.

C. Backmapping

The current expTM model can only generate data in CV space. To map generated CVs (λ_1, λ_3) back to actual atomic configurations, we use a nearest-neighbor approach: specifically, we run through all possible pairs between the MD-simulated data and the expTM-generated CVs to find the smallest norm difference (Supplementary Fig. 9). From each reference pressure $(P = 1\,\text{GPa}$ and $P = 8\,\text{GPa})$, the generated points correctly match the characteristic structures observed in the real system (i.e., an alternating arrangement at low pressure and a linear arrangement at high pressure).



Supplementary Fig. 8: Collective variable values (λ_1, λ_3) for the configurations used to train expTM. Each point represents the mean molecule-wise order parameters.



Supplementary Fig. 9: Snapshots of CO_2 from MD simulations (top) and generated configurations via backmapping (bottom).

[1] I. Gimondi and M. Salvalaglio, Co2 packing polymorphism under pressure: Mechanism and thermodynamics of the i-iii polymorphic transition, The Journal of chemical physics **147** (2017).

[2] M. J. Abraham, T. Murtola, R. Schulz, S. Páll, J. C. Smith, B. Hess, and E. Lindahl, Gromacs: High performance molecular simulations through multi-level parallelism from laptops to supercomputers, SoftwareX **1**, 19 (2015).

[3] S. Páll, M. J. Abraham, C. Kutzner, B. Hess, and E. Lindahl, Tackling exascale software challenges in molecular dynamics simulations with gromacs, in *Solving Software Challenges for Exascale: International Conference on Exascale Applications and Software, EASC 2014, Stockholm, Sweden, April 2-3, 2014, Revised Selected Papers 2* (Springer, 2015) pp. 3–27.

[4] G. Bussi, D. Donadio, and M. Parrinello, Canonical sampling through velocity rescaling, The Journal of chemical physics **126** (2007).

[5] H. J. Berendsen, J. v. Postma, W. F. Van Gunsteren, A. DiNola, and J. R. Haak, Molecular dynamics with coupling to an external bath, The Journal of chemical physics **81**, 3684 (1984).

[6] B. Hess, P-lincs: A parallel linear constraint solver for molecular simulation, Journal of chemical theory and computation **4**, 116 (2008).

[7] G. A. Tribello, M. Bonomi, D. Branduardi, C. Camilloni, and G. Bussi, Plumed 2: New feathers for an old bird, Computer physics communications **185**, 604 (2014).

[8] A. Barducci, G. Bussi, and M. Parrinello, Well-tempered metadynamics: a smoothly converging and tunable free-energy method, Physical review letters **100**, 020603 (2008).

[9] M. Salvalaglio, C. Perego, F. Giberti, M. Mazzotti, and M. Parrinello, Molecular-dynamics simulations of urea nucleation from aqueous solution, Proceedings of the National Academy of Sciences **112**, E6 (2015).