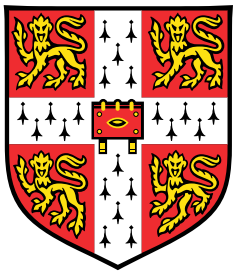
Transferable Neural Transports for Parallel Tempering



Antonio Franca

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Master of Philosophy in Machine Learning and Machine Intelligence

Queens' College August 2025

To love, curiosity, and the simple miracle of being alive.

Declaration

I, Antonio Franca of the Queens College, candidate for the *MPhil in Machine Learning and Machine Intelligence*, hereby declare that this dissertation is my own work, unaided except where acknowledged, and contains no material previously submitted for any comparable purpose. This work utilises the Microsoft Timewarp dataset (https://huggingface.co/datasets/microsoft/timewarp) and adapts components from the MIT–licensed repositories microsoft/-timewarp, VincentStimper/boltzmann-generators, and the open-source implementation PTSD. The code accompanying this thesis is available at this `link`.

Antonio Franca
August 2025

Acknowledgements

First and foremost, I would like to thank my supervisors, José Miguel Hernández-Lobato, Tony RuiKang OuYang, and Jiajun He, for introducing me to the fascinating world of molecules and their intricacies, and for their guidance and support. I am also deeply grateful to everyone I met in Cambridge, especially my fellow master's classmates, for the engaging conversations we shared; I will always cherish having crossed paths with you. Finally, I would like to thank my parents, my girlfriend Victoria, and all my friends for their love and support. Wishing you all the very best.

Abstract

Parallel tempering (PT) accelerates sampling by swapping configurations between temperatures, yet swap acceptance collapses as molecular dimension grows because adjacent tempered targets have vanishing overlap. We address this by learning *bijective transports* that *construct* overlap between adjacent temperatures while preserving exactness. Concretely, we use an involutive deterministic swap with a Jacobian-corrected Metropolis rule that maintains detailed balance, and we show that a bidirectional likelihood objective aligns with the swap-acceptance exponent, directly training transports to increase acceptance. We propose three model families along the bias–capacity spectrum: a coordinate-only RealNVP (*PTSwapFlow*), a symmetry-aware graph flow (*PTSwapGraphFlow*), and a global self-attention model with auxiliary variables (*PTSwapTransformerFlow*). On a nine-dipeptide dataset spanning 300–1000 K, vanilla PT exhibits uniformly low, size-dependent swap acceptance, whereas our learned transports consistently boost acceptance across systems, including held-out (unseen) peptides, providing initial evidence of transferability.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

The first principle is that you must not fool yourself—
and you are the easiest person to fool.
— Richard P. Feynman (1918–1988)

This thesis addresses a fundamental problem in computational statistics and molecular simulation: how to draw representative samples from a high–dimensional target distribution known only up to a normalizing constant. Concretely, for a configuration space $\Omega \subset \mathbb{R}^{3N}$ and an energy function $U : \Omega \to \mathbb{R}$, the Boltzmann law

$$\mu_\beta(d\mathbf{x}) \;=\; Z_\beta^{-1} \exp\big(-\beta\, U(\mathbf{x})\big)\, d\mathbf{x}, \qquad Z_\beta = \int_\Omega \exp\big(-\beta\, U(\mathbf{x})\big)\, d\mathbf{x},$$

governs the equilibrium statistics of molecular conformations at inverse temperature $\beta = (k_B T)^{-1}$ (Boltzmann, 1868; Risken, 1996). Direct evaluation of Z_β is intractable in all but toy systems, so *sampling* becomes the principal computational task. This "unnormalized density" setting pervades Bayesian inference, statistical mechanics, and modern generative modeling.

Two complementary paradigms dominate this landscape. The first designs *Markov kernels* whose invariant measure is the target and that mix rapidly enough to explore it well. Metropolis–Hastings (MH), Metropolis-Adjusted Langevin Algorithm (MALA), and Hamiltonian Monte Carlo (HMC) exemplify this approach: they propose local or gradient–informed moves and correct with an acceptance test that cancels the unknown partition function (Douc and Cor, 2015; Hamilton, 1834). While powerful, local dynamics face an *exponential barrier* problem (detailed in Chapter 2): expected barrier–crossing times scale like $\exp(\beta\,\Delta U)$, rendering rare events prohibitively expensive in high dimensions (Kramers, 1940). *Parallel tempering* (PT) mitigates this by coupling multiple replicas at different temperatures;

high–temperature chains explore broadly and share information with low–temperature chains via replica–exchange moves (Predescu et al., 2004). However, PT itself suffers from the *temperature–overlap problem* (detailed in Chapter 3): adjacent tempered distributions overlap weakly as system size grows, driving swap acceptance down and the required number of replicas up (Predescu et al., 2004).

The second paradigm builds *transport maps* that push a simple source distribution toward the target. Normalizing flows parameterize invertible maps with tractable Jacobians; likelihoods are exact and sampling is direct (Dinh et al., 2017; Kingma and Dhariwal, 2018; Papamakarios et al., 2018; Behrmann et al., 2019). Diffusion models and flow matching relax strict invertibility to gain expressivity, learning vector fields or score functions that connect noise to data (Ho et al., 2020; Chen et al., 2019; Grathwohl et al., 2018; Lipman et al., 2023). In molecular contexts, flows have been shown to accelerate sampling and to transfer across related systems when architectures and training pipelines are chosen carefully (Klein, Foong, et al., 2023; Tan, Bose, et al., 2025; Klein and Noé, 2024).

These paradigms need not compete. Recent work has highlighted a constructive synergy: *learned transports* can be embedded into PT to *manufacture* overlap between neighboring tempered targets, replacing naive coordinate swaps with *flow–aligned* swaps that enjoy much higher acceptance while preserving exactness via Metropolis corrections (Zhang et al., 2025a). At a high level, PT supplies global connectivity across temperatures, and the transport supplies local efficiency by aligning the pairwise marginals.

Our focus is equilibrium sampling of small peptides in Cartesian coordinates. Probabilistic modeling of conformational ensembles enables binding and solvation free–energy estimation, uncertainty–aware structure prediction, and comparison to ensemble–averaged experimental observables, all tasks that depend on integrals under μ_β rather than on following real–time dynamics (Risken, 1996). While Langevin or Moleculat Dynamics (MD) integrators target the correct invariant law (Langevin, 1908; Risken, 1996), they waste effort on femtosecond vibrations; equilibrium samplers that jump directly between conformations can be far more compute–efficient in this regime. The central technical challenge, then, is to design neural transports that (i) are expressive enough to align adjacent tempered Boltzmann measures, (ii) respect molecular symmetries where possible, and (iii) transferable across peptides.

1.1 Contributions

This thesis is contributed as follows:

- We reproduce accelerated parallel tempering (PT) via learned transports work (Zhang et al., 2025b) implementing an *involutive* deterministic swap with exact Metropolis correction.
- We propose two *transferable* model families that work at cross–peptide generalization: a *graph–aware* flow, and a *transformer-based* flow.
- We analyze permutation and $E(3)$ properties in the proposed models, both theoretically and empirically.
- We derive the proper augmented negative log-likelihood loss (NLL) and acceptance exponent in the augmented normalizing flow borrowed from (Klein, Foong, et al., 2023).

1.2 Outline

The thesis is organized as follows:

- Chapter 2 introduces equilibrium sampling for molecular systems and reviews Markov-chain Monte-Carlo (MCMC) kernels, parallel tempering (PT), and neural transports (flows, diffusion, flow matching) alongside graph neural networks (GNNs).
- Chapter 3 formalizes flow–aligned replica swaps, presents our three architectures (coordinate, graph, transformer), learning objectives, and symmetry analysis.
- Chapter 4 details datasets, evaluates swap acceptance, transfer to unseen peptides; and reports results on symmetry remedies, and other physical validation metrics.
- Chapter 5 summarizes findings and limitations and outlines future work.

Chapter 2

Background

The computer is incredibly fast, accurate, and stupid.
Man is unbelievably slow, inaccurate, and brilliant.
Together they are powerful beyond imagination.
— Leo Cherne (1912–1999)

In this chapter we outline the necessary background, which spans multiple topics. Our aim is to tie them together in a cohesive narrative. To begin, we introduce statistical-mechanical motivation for equilibrium sampling of molecular conformations. We then survey classical Markov-chain Monte Carlo (MCMC) kernels. To overcome the barrier-crossing limits of these local kernels, we turn to parallel tempering (PT), and the use of neural transports to speed up replica exchanges and mixing. Finally, we introduce graph neural networks (GNNs) to integrate molecular information into neural models.

2.1 Why sampling molecular conformations?

The conformation of an N-atom molecule can be fully specified by its Cartesian coordinates in three-dimensional space

$$\mathbf{x} = (\mathbf{r}_1, \ldots, \mathbf{r}_N) \in \Omega \subset \mathbb{R}^{3N},$$

where $\mathbf{r}_i = (x_i, y_i, z_i)^\top$ is the three–dimensional position of atom i, and Ω denotes the region of configuration space allowed by hard-sphere repulsion, bond-length constraints, and any other natural laws our reality imposes. On this $3N$-dimensional manifold we define a potential-energy surface $U : \Omega \to \mathbb{R}$ whose gradients generate the familiar Newtonian flow

$$m_i \ddot{\mathbf{r}}_i(t) = -\nabla_{\mathbf{r}_i} U(\mathbf{r}_1(t), \ldots, \mathbf{r}_N(t)), \quad i = 1, \ldots, N. \tag{2.1}$$

This second-order ordinary differential equation (ODE) underlies molecular dynamics (MD) simulations. Intuitively, one passes from this deterministic MD trajectory in Equation (2.1) to a probability law by *randomising the starting point*. Imagine launching many replicas of the molecule, each initialised with tiny, independent perturbations that reflect the ambient temperature. If the system is *ergodic*, meaning every region of configuration space is eventually visited, then the long-time histogram of atomic positions built from this swarm will stabilise. Classical results in statistical mechanics (Boltzmann, 1868) show that the limiting frequency of any conformation $\mathbf{x}$ is proportional to the Boltzmann weight

$$\mu_\beta(d\mathbf{x}) = \frac{1}{Z_\beta} e^{-\beta U(\mathbf{x})}\, \lambda(d\mathbf{x}), \quad Z_\beta = \int_\Omega e^{-\beta U(\mathbf{x})}\, d\lambda(\mathbf{x}), \tag{2.2}$$

where $\beta = (k_B T)^{-1}$, with k_B denoting the Boltzmann constant, T the absolute temperature of the heat bath, and λ is the Lebesgue measure on $\mathbb{R}^{3N}$. Another, more systematic way to view MD as probabilistic sampling is to embed the randomness directly into Equation (2.1) via Langevin dynamics. Physically, this mimics a molecule diffusing in a solvent at temperature T, i.e. molecular collisions impart random kicks, while viscous drag removes momentum (Langevin, 1908). Mathematically, one augments Newton's Equation (2.1) with a linear damping term and Gaussian noise, yielding the underdamped stochastic differential equation (SDE)

$$\begin{aligned} d\mathbf{r}_i &= \frac{\mathbf{p}_i}{m_i}\, dt, \\ d\mathbf{p}_i &= -\nabla_{\mathbf{r}_i} U(\mathbf{x})\, dt - \gamma \mathbf{p}_i\, dt + \sqrt{2\gamma m_i/\beta}\; d\mathbf{W}_i(t), \end{aligned} \tag{2.3}$$

where $\gamma > 0$ is a friction (collision) coefficient and $d\mathbf{W}_i(t)$ is a standard Wiener process (Gaussian noise) term. The first term in $d\mathbf{p}_i$ is the deterministic force as in Newton's law; the second and third terms represent Stokes' drag and random hits from the heat bath. By taking the high-friction limit $\gamma \to \infty$ (with $D_i = k_B T/(\gamma m_i)$ fixed), momenta decorrelate and one recovers the overdamped SDE,

$$d\mathbf{r}_i = -D_i \beta\, \nabla_{\mathbf{r}_i} U(\mathbf{x})\, dt \; + \; \sqrt{2D_i}\, d\mathbf{W}_i(t). \tag{2.4}$$

Crucially, one shows via the Fokker–Planck argument (Risken, 1996) that both Equation (2.3) and its overdamped limit in Equation (2.4) share the same unique invariant measure, namely the Boltzmann density μ_β from Equation (3.1). Yet, faithfully integrating either Equation (2.1) or its Langevin variants forces our timestep down to the fastest bond-stretch and angle-bend vibrations, on the order of 1–2 femtoseconds (10^{-15} s). Thus, to reach biolog-

ically relevant timescales, such as the millisecond (10^{-3} s) folding of a small protein, we would need on the order of 10^{12} steps, making such simulations prohibitively expensive. However, the femtosecond-scale is only relevant if we care about the detailed dynamical trajectory, but this simulation barely captures the long-time occupancy of conformations, exactly the information we need when computing binding free energies, solvation or partition free energies, or ensemble-averaged NMR observables.

Since any ergodic thermostat (e.g. underdamped Langevin) converges to the same Boltzmann law, we can avoid resolving every tiny oscillation when our aim is merely equilibrium averages. Instead, by leaping directly between conformations and sampling μ_β, we concentrate compute where the probability mass lies. This insight, known as *equilibrium sampling*, drives the focus of this thesis: designing a transferable, ML-driven sampler that yields high-quality Boltzmann samples even for unseen peptides. To conclude this motivational overview, we recall then this Boltzmann Equation (3.1) assigns each molecular conformation its equilibrium probability, underpinning calculations of equilibrium observables. In particular, any equilibrium property can be written as

$$\langle f \rangle_\beta = \int_\Omega f(\mathbf{x})\, \mu_\beta(d\mathbf{x}) \tag{2.5}$$

where $f : \Omega \to \mathbb{R}$ is measurable. Unfortunately, direct integration of Equation (2.5) is infeasible because Ω has very high dimension (typically $3N$ with $N \sim 10^2$ or more), and Z_β cannot be computed analytically. As a result, we must rely on sampling methods to approximate $\langle f \rangle_\beta$. Once high-quality samples $x^{(i)} \sim \mu_\beta$ are available, any observable $\langle f \rangle_\beta$ can be approximated using the usual Monte Carlo estimator. In the next section, we introduce some classical sampling techniques that generate draws from *unnormalized* densities, such as Equation (3.1).

2.2 Classical MCMC Methods

Having motivated *why* one wishes to draw equilibrium conformations, we now turn to *how*. In the previous section we argued that MD, while physically faithful, wastes effort by resolving femtosecond vibrations when our real interest is *equilibrium statistics*. Markov–chain Monte-Carlo (MCMC) replaces the physical trajectory by a *fictitious* random walk in configuration space that is designed to visit conformations with exactly the Boltzmann frequency, yet can jump over unimportant regions far more quickly than MD.

2.2.1 Markov-Chain Formalism

A Markov chain on the configuration space Ω is fully specified by its *transition kernel*.

Definition (Transition kernel). Let Ω be the configuration space equipped with its Borel σ-algebra $\mathscr{B}(\Omega)$. A sequence of random variables $(X_t)_{t\in\mathbb{N}}$ taking values in Ω is called a *Markov chain* with *transition kernel* $K\colon \Omega\times\mathscr{B}(\Omega)\to[0,1]$ if, for every $t\geq 0$ and every measurable set $A\subseteq\Omega$,

$$\Pr\big(X_{t+1}\in A \;\big|\; X_t = x, X_{t-1},\dots,X_0\big) \;=\; K(x,A).$$

Once K is specified, the sampling problem reduces to running the chain long enough so that its distribution converges to the target μ_β. To make this precise we need the notion of an invariant measure.

Definition (Invariant measure). A probability measure π on $(\Omega,\mathscr{B}(\Omega))$ is said to be *invariant* under the kernel K if

$$\int_\Omega \pi(dx)\,K(x,A) \;=\; \pi(A), \qquad \forall A\in\mathscr{B}(\Omega). \tag{2.6}$$

Invariance implies then that, if $X_t\sim\pi$, then also $X_{t+1}\sim\pi$: the chain leaves π unchanged. A sufficient condition for invariance is *detailed balance*

$$\mu_\beta(dx)\,K(x,dy) \;=\; \mu_\beta(dy)\,K(y,dx). \tag{2.7}$$

However, while detailed balance guarantees that π is a stationary measure for K, it by itself does not ensure that the chain will actually reach that equilibrium from an arbitrary start. To guarantee convergence to μ_β from any initial X_0, the chain must be irreducible and aperiodic. Together these properties comprise *ergodicity*, which by standard theorems (Aldous and Fill, 2002) implies

$$\mathscr{L}(X_t) \xrightarrow[t\to\infty]{} \mu_\beta \quad\text{and}\quad \frac{1}{T}\sum_{t=1}^{T} f(X_t) \xrightarrow{\text{a.s.}} \int f(x)\,\mu_\beta(dx),$$

meaning that (i) the distribution of X_t converges to the Boltzmann law, and (ii) time-averages along the chain recover true ensemble averages of any observable f.

By formalizing it this way, we see that any MCMC sampling algorithm differs only in its choice of the kernel K. The challenge becomes designing K to mix rapidly (i.e. minimise autocorrelation times) while preserving invariance (2.6).

2.2.2 Local Kernels: MH, MALA, HMC

Having introduced some necessary formalisms, we now describe how to build a kernel K that satisfies detailed balance for the Boltzmann target $\pi(\mathbf{x}) \propto e^{-\beta U(\mathbf{x})}$. For this, we generally make use of the Metropolis–Hastings (MH) procedure which corrects the kernel proposal so that assymptotically our sampling mechanism converges to the target distribution.

Definition (MH kernel). MH proceeds in two stages. Given the current state $\mathbf{x}$,

$$\boxed{\text{(i) propose } \mathbf{y} \sim q(\mathbf{y} \mid \mathbf{x}) \quad \Longrightarrow \quad \text{(ii) accept with } \alpha(\mathbf{x},\mathbf{y}) = \min\Big\{1, r(\mathbf{x},\mathbf{y})\Big\}} \tag{2.8}$$

where q is any convenient transition density, π is an arbitrary target distribution from which we want to obtain samples, and the acceptance ratio $r(\mathbf{x},\mathbf{y}) = \frac{\pi(\mathbf{y})q(\mathbf{x}|\mathbf{y})}{\pi(\mathbf{x})q(\mathbf{y}|\mathbf{x})}$

In our Boltzmann sampling scenario, the acceptance ratio $r(\mathbf{x},\mathbf{y})$ involves the *difference* of potential energies $U(\mathbf{y}) - U(\mathbf{x})$, and does not require knowledge of the partition function Z_β. For symmetric kernel proposals, moves that decrease energy ($U(\mathbf{y}) < U(\mathbf{x})$) are always accepted, while moves that increase energy are accepted in proportion to Boltzmann probability $\exp[-\beta\,\Delta U]$, exactly compensating for their lower target density.

The trade-off presented is that a small proposal step δ yields high acceptance but slow, diffusive exploration. A large δ explores aggressively but is often rejected. By choosing q to propose larger or more informed steps (e.g. incorporating energy gradients), one influences the mixing speed of the chain, and the invariance of target density π_β is guaranteed by the accept/reject mechanism.

Remark. One verifies by simple algebra that the resulting transition kernel (see A.1)

$$K(\mathbf{x}, d\mathbf{y}) = q(\mathbf{y} \mid \mathbf{x})\,\alpha(\mathbf{x},\mathbf{y})\,d\mathbf{y} \;+\; \Big[1 - A(\mathbf{x})\Big]\,\delta_{\mathbf{x}}(d\mathbf{y}), \tag{2.9}$$

with $A(\mathbf{x}) = \int q(\mathbf{y} \mid \mathbf{x})\,\alpha(\mathbf{x},\mathbf{y})\,d\mathbf{y}$, satisfies detailed balance (2.7).

Thus, by embedding the target π_β into the design of the transition kernel K and enforcing detailed balance, we obtain a Markov chain whose long-time behavior correctly samples an arbitrary unnormalized target (such as the Boltzmann density) without ever computing the intractable normalizing constant. The only remaining incognita at this point is what proposal distribution $q(\mathbf{y} \mid \mathbf{x})$ should we use to explore the density π_β as *efficiently as possible*, i.e. one that rapidly yields representative samples so accurate estimates can be obtained from few draws (high effective sample size per unit compute)."

A naive choice is to pick a simple, state-dependent Gaussian proposal $q(\mathbf{y} \mid \mathbf{x}) = \mathcal{N}\big(\mathbf{y}; \mathbf{x}, \sigma^2 I_d\big)$ and apply the plain MH accept–reject criteria. While this *random-walk MH* can work in low dimensions, it quickly succumbs to the *curse of dimensionality* (Bellman, 1957). However, why move blindly when the energy gradient points where the probability mass lies. A natural improvement is the Metropolis–adjusted Langevin algorithm (MALA), which seeds each proposal with one Euler–Maruyama step of the overdamped Langevin SDE.

Definition (MALA proposal). The Metropolis-Adjusted Langevin Algorithm (MALA) proposes moves via

$$\mathbf{y} = \underbrace{\mathbf{x} - \tfrac{\delta^2}{2}\nabla U(\mathbf{x})}_{\text{drift}} + \underbrace{\delta\,\xi}_{\text{noise}}, \qquad \xi \sim \mathcal{N}(0, I),$$

where the drift term biases proposals toward lower energy regions, and the noise term injects isotropic Gaussian randomness to ensure proper exploration of configuration space.

Naturally, a final MH correction restores exact sampling from the target distribution. The choice of MALA proposal distribution is then $q(\mathbf{y} \mid \mathbf{x}) = \mathcal{N}\Big(\mathbf{y}; \mathbf{x} - \tfrac{\delta^2}{2}\nabla U(\mathbf{x}), \delta^2 I\Big)$. Yet MALA remains fundamentally local and can be blocked by high energy barriers. Hamiltonian Monte Carlo (HMC) pretends to overcome this by introducing auxiliary momentum and simulating Hamilton's equations.

Definition (HMC proposal). Hamiltonian Monte Carlo (HMC) constructs a proposal by first sampling an auxiliary momentum $\mathbf{p} \sim \mathcal{N}(\mathbf{0}, M)$, and then evolving the pair $(\mathbf{x}, \mathbf{p})$ under Hamilton's equations (Hamilton, 1834)

$$\dot{\mathbf{x}} = M^{-1}\mathbf{p}, \qquad \dot{\mathbf{p}} = -\nabla U(\mathbf{x})$$

for L steps of size ε using a reversible, volume-preserving leapfrog integrator. The new position $\mathbf{x}'$ (with refreshed momentum) is taken as the proposal.

Again, a Metropolis–Hastings correction restores exact sampling. We conclude this subsection after examining two key MH-based kernels driven by energy gradients; MALA and HMC. Although these methods dramatically outperform blind random walks, they remain fundamentally local because of what we might consider to be the *exponential barrier problem* (Kramers, 1940).

Exponential Barrier Problem. For a metastable basin separated by a barrier of height ΔU, the mean first-passage (exit) time under *single-temperature, local* dynamics (e.g., MALA, HMC) follows an Arrhenius law

$$\mathbb{E}[\tau_{\text{exit}}] \asymp C \exp(\beta \Delta U),$$

where the prefactor C depends on curvatures, friction, and proposal details. Gradient information or momentum can improve C but not the exponential dependence on $\beta\Delta U$. Consequently, barrier crossing is exponentially rare.

2.2.3 Parallel Tempering

Because of the *exponential-barrier problem*, local MCMC kernels make only small, neighborhood-limited moves and can become trapped in a single energy basin, rarely transitioning to others. In molecular applications, these basins are *metastable conformational states* (e.g., distinct folded, unfolded, or intermediate protein structures). Tempering–based ensemble samplers mitigate this by running multiple replicas at distinct inverse temperatures and allowing them to exchange information. We index inverse temperatures as

$$\beta_1 = \beta \; > \; \beta_2 \; > \; \cdots \; > \; \beta_m \approx 0,$$

where $\beta_i = (k_{\mathrm{B}} T_i)^{-1}$ controls the sharpness of the Boltzmann weight $e^{-\beta_i U}$. Small β_i (high T_i) flatten the landscape and facilitate barrier crossing; large β_i (low T_i) concentrate probability near minima.

Definition (Replica ladder). Fix an ordered set $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_m)$ with $\beta_1 = \beta$ and $0 < \beta_{i+1} < \beta_i$. For each i, define the tempered Boltzmann measure

$$\mu_{\beta_i}(d\mathbf{x}) \; = \; \frac{1}{Z_{\beta_i}} e^{-\beta_i U(\mathbf{x})} \, \lambda(d\mathbf{x}), \qquad Z_{\beta_i} = \int_{\Omega} e^{-\beta_i U(\mathbf{x})} \, d\lambda(\mathbf{x}). \tag{2.10}$$

At the highest temperature ($\beta_m \approx 0$), μ_{β_m} approaches the base measure λ, enabling broad exploration. The ensemble chain targets the product

$$\boldsymbol{\mu}(d\mathbf{x}_1, \ldots, d\mathbf{x}_m) \; = \; \prod_{i=1}^{m} \mu_{\beta_i}(d\mathbf{x}_i),$$

and observables are computed from the physical-temperature component $\mathbf{x}_1$. The PT algorithm alternates between two phases:

Local sampling. Each replica evolves independently for τ steps using any kernel K_i that preserves μ_{β_i} (typically MALA or HMC). Because these updates act on disjoint coordinates, the product kernel $K_{\text{local}} = \bigotimes_{i=1}^{m} K_i$ preserves $\boldsymbol{\mu}$ and inherits detailed balance.

Replica exchange. After the local updates, select an adjacent pair $(i, i+1)$ (e.g., cycle through neighbors or choose with probabilities $\rho_{i,i+1}$) and propose the swap

$$(\mathbf{x}_i, \mathbf{x}_{i+1}) \longrightarrow (\mathbf{x}_{i+1}, \mathbf{x}_i). \tag{2.11}$$

Then, accept this swap with probability

$$\alpha_{\text{swap}} = \min\Big\{1, \exp\big[(\beta_i - \beta_{i+1})\big(U(\mathbf{x}_{i+1}) - U(\mathbf{x}_i)\big)\big]\Big\}, \tag{2.12}$$

which is exactly the Metropolis–Hastings rule (2.8) applied to the two–replica target $\mu_{\beta_i} \otimes \mu_{\beta_{i+1}}$ with the deterministic swap proposal from Equation (3.3). We make this precise in Chapter 3. This Metropolis criterion enforces detailed balance (see Appendix A.1 for a derivation for arbitrary targets) for the ensemble while allowing high-temperature replicas to help low-temperature ones overcome energy barriers.

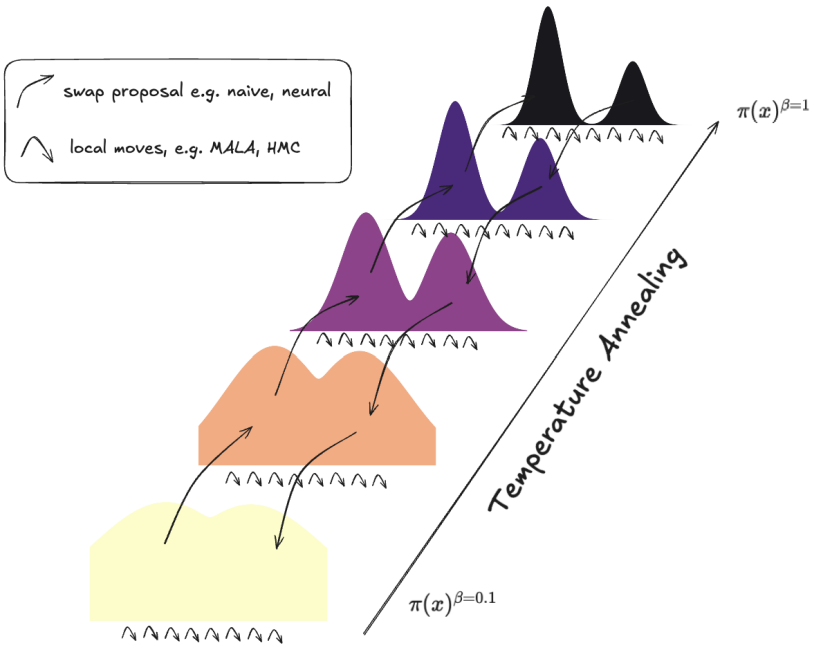


Fig. 2.1 Temperature-ladder ensemble sampler. Each replica (colored curves) explores a tempered density via local moves (small wiggles) and occasionally proposes swaps between adjacent temperatures (arrows).

In summary, coupling a spectrum of temperatures via swaps turns rare barrier crossings at low temperature into ordinary moves in the expanded space Ω^m. Each replica samples its own Boltzmann distribution, and occasional exchanges let the target-temperature replica inherit high-temperature tunneling moves. There is, however, no free lunch: tempering requires simulating M replicas (more computationally expensive) and suffers from the *temperature overlap problem*: in high dimensions, neighboring tempered distributions overlap weakly, causing low swap acceptance and necessitating many replicas. (Predescu et al., 2004). We make this more precise in Chapter 3.

2.3 Neural Transports

The *temperature–overlap problem* in PT is a severe algorithmic bottleneck: acceptable swap rates require substantial overlap between adjacent tempered distributions, yet in high–dimensional systems their natural overlap shrinks rapidly with system size. The transport paradigm offers a constructive remedy: rather than relying on incidental overlap, we *learn* maps that align neighboring tempered targets, creating overlap by design and enabling high–acceptance swaps. Recent advances (Zhang et al., 2025b) show that neural samplers—e.g., normalizing flows and diffusion models—can substantially reduce the number of replicas and the wall–clock cost of PT. This section develops the theoretical basis for this approach.

2.3.1 Transport Paradigm

A central problem in computational statistics is to transform samples from one probability distribution into another. Given probability measures π and ρ on a common measurable space $(\Omega, \mathscr{F})$, the goal is to find a measurable map $T : \Omega \to \Omega$ such that, if $\mathbf{x} \sim \pi$, then $T(\mathbf{x})$ is (approximately) distributed as ρ. Equivalently, we seek a map for which the pushforward measure

$$T_{\#}\pi(A) \;:=\; \pi\big(T^{-1}(A)\big), \qquad A \in \mathscr{F},$$

is close to ρ under some discrepancy $\mathscr{D}(T_{\#}\pi, \rho)$, e.g., Kullback–Leibler divergence, Wasserstein distance, or maximum mean discrepancy. Methods for constructing T typically trade off *expressivity* (how rich a family of transformations is allowed) against *tractability* (whether densities, Jacobians, inverses, or samples are easy to compute).

Definition (Neural transport map). A *neural transport* is a parameterized map $T_\theta : \Omega \to \Omega$ (with parameters θ, typically neural-network weights) learned so that the pushforward $T_{\theta\#}\pi$ approximates a target ρ by minimizing a chosen discrepancy $\mathscr{D}(T_{\theta\#}\pi, \rho)$.

In parallel tempering, we employ transport maps between adjacent tempered distributions,

$$T_\theta^{(k)} : \mu_{\beta_k} \longrightarrow \mu_{\beta_{k+1}},$$

so that proposals aligned by $T_\theta^{(k)}$ achieve high acceptance and alleviate the usual overlap bottleneck. Three broad classes of neural transports instantiate different points on the expressivity–tractability spectrum: *normalizing flows* (invertible maps with tractable Jacobians), *diffusion models* (score-based generative samplers), and *flow matching* (deterministic ODE transports). In this thesis we focus on normalizing flows as the core transport architecture, and subsequently provide a brief overview of diffusion- and flow-matching–based transports to show how the same swap-and-transport framework extends naturally.

2.3.2 Normalizing Flows

Definition (Normalizing flow). A *normalizing flow* is a bijection

$$T_\theta = T_\theta^{(K)} \circ \cdots \circ T_\theta^{(1)} : \mathbb{R}^d \to \mathbb{R}^d,$$

built as a composition of K invertible layers, each with tractable inverse and Jacobian determinant. Given a base density p_0 on $\mathbb{R}^d$, the flow induces the target density p_θ as the pushforward $T_{\theta\#}p_0$.

The compositional design is crucial: rather than learning a single monolithic bijection, one stacks simple, easily invertible layers. Each layer effects a small deformation with cheap log–determinant computation; their composition can represent highly non-linear and multimodal transports while retaining numerical stability and tractability.

Definition (Change of variables). For a bijection $T : \mathbb{R}^d \to \mathbb{R}^d$ with inverse T^{-1}, if $\mathbf{y} = T(\mathbf{x})$ then

$$p_Y(\mathbf{y}) = p_X\big(T^{-1}(\mathbf{y})\big)\,\big|\det \nabla T^{-1}(\mathbf{y})\big|.$$

Equivalently,

$$\log p_Y(\mathbf{y}) = \log p_X\big(T^{-1}(\mathbf{y})\big) + \log\big|\det \nabla T^{-1}(\mathbf{y})\big|.$$

Applied layer–by–layer, this yields

$$\log p_\theta(\mathbf{x}) = \log p_0\big(T_\theta^{-1}(\mathbf{x})\big) - \sum_{k=1}^{K} \log\left|\det \nabla T_\theta^{(k)}(\mathbf{z}_{k-1})\right|, \qquad \mathbf{z}_0 = T_\theta^{-1}(\mathbf{x}),\ \mathbf{z}_k = T_\theta^{(k)}(\mathbf{z}_{k-1}).$$

In our PT setting we will use $p_0 = \mu_{\beta_k}$ and aim for $p_\theta \approx \mu_{\beta_{k+1}}$, training $T_\theta^{(k)}$ to minimize a discrepancy $\mathscr{D}\Big(T_{\theta\#}^{(k)}\mu_{\beta_k}, \mu_{\beta_{k+1}}\Big)$.

A flow is specified by its invertible layers $T_\theta^{(k)}$. The only design choice is then the form of them. Common, tractable choices include:

(i) Affine coupling layers (Dinh et al., 2017; Kingma and Dhariwal, 2018). Split $x = (x_a, x_b)$ and update

$$y_a = x_a, \qquad y_b = x_b \odot \exp s_\theta(x_a) + t_\theta(x_a).$$

The Jacobian is block–triangular, so $\log|\det J| = \sum_j s_\theta(x_a)_j$ is $\mathscr{O}(d)$, and inversion is immediate. Interleaving permutations ensures all coordinates are eventually transformed.

(ii) Autoregressive layers (Papamakarios et al., 2018; Kingma, Salimans, et al., 2017). Update components sequentially,

$$y_i = \mu_\theta(x_{1:i-1}) + \sigma_\theta(x_{1:i-1})\, x_i, \quad i = 1, \ldots, d.$$

The Jacobian is triangular, giving analytic log–det (sum of diagonals). In MAF, density evaluation is fast (parallel) while sampling is sequential; in IAF, sampling is fast (parallel) while exact density evaluation is costly.

(iii) Residual and continuous–time layers (Behrmann et al., 2019; Chen et al., 2019; Grathwohl et al., 2018). Use $y = x + u_\theta(x)$. If u_θ is contractive (e.g., $\|\nabla u_\theta\| < 1$), invertibility follows from the Banach fixed–point theorem (Banach, 1922) and

$$\log\det(I + \nabla u_\theta) = \sum_{k\geq 1} \frac{(-1)^{k+1}}{k} \operatorname{Tr}\Big[(\nabla u_\theta)^k\Big],$$

estimated via stochastic trace estimators. In the limit of infinitesimal layers one obtains a continuous normalizing flow (CNF) governed by the neural ODE

$$\frac{dx_t}{dt} = f_\theta(x_t, t), \qquad \frac{d}{dt}\log p_t(x_t) = -\operatorname{Tr}[\nabla_{x_t} f_\theta(x_t, t)]\,.$$

In this thesis we employ affine coupling flows due to their linear-time sampling and log-density evaluation, and because they have performed well on closely related molecular

problems (e.g., (Klein, Foong, et al., 2023; Dinh et al., 2017)). Brief overviews of diffusion- and flow-matching–based transports follow to illustrate how our swap-and-transport framework extends beyond flows.

2.3.3 Diffusion Models

Diffusion models generate samples by learning the reverse of a simple *noising* process that progressively corrupts data. They trade the strict invertibility of normalizing flows for greater expressivity in high dimensions and excellent sample quality.

Definition (Diffusion process). A diffusion model comprises (i) a *forward* process that adds noise to data and (ii) a *reverse* process that learns to denoise by estimating the score $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$ of the forward marginals. In continuous time, a common choice for the forward process is the variance–preserving (VP) SDE

$$d\mathbf{x}_t \;=\; -\tfrac{1}{2}\beta(t)\,\mathbf{x}_t\,dt \;+\; \sqrt{\beta(t)}\,d\mathbf{W}_t,$$

where $\beta(t) > 0$ is a noise schedule and $\mathbf{W}_t$ is standard Brownian motion. This SDE admits tractable Gaussian conditionals $q(\mathbf{x}_t \mid \mathbf{x}_0)$, enabling direct sampling at any time t.

A fundamental result (reverse–time diffusion) gives the *reverse* SDE

$$d\mathbf{x}_t \;=\; \big[-\tfrac{1}{2}\beta(t)\,\mathbf{x}_t \;-\; \beta(t)\,\nabla_{\mathbf{x}} \log p_t(\mathbf{x}_t)\big]\,dt \;+\; \sqrt{\beta(t)}\,d\bar{\mathbf{W}}_t,$$

which evolves from noise back to data. The only unknown is the score $\nabla_{\mathbf{x}} \log p_t(\mathbf{x})$; diffusion models approximate it with a neural network $s_\theta(\mathbf{x}, t) \approx \nabla_{\mathbf{x}} \log p_t(\mathbf{x})$.

In practice one samples $(t, \mathbf{x}_0, \boldsymbol{\varepsilon})$ and forms a noised datum

$$\mathbf{x}_t \;=\; \sqrt{\bar{\alpha}_t}\,\mathbf{x}_0 \;+\; \sqrt{1-\bar{\alpha}_t}\,\boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim \mathcal{N}(0, I),$$

where $\bar{\alpha}_t \in (0, 1]$ encodes the noise schedule (discrete DDPM (Ho et al., 2020)) or its continuous analogue (VP SDE). Since $q(\mathbf{x}_t \mid \mathbf{x}_0)$ is Gaussian, its score is known in closed form, and one minimizes a weighted denoising score–matching objective

$$\mathscr{L}(\theta) \;=\; \mathbb{E}_{t, \mathbf{x}_0, \boldsymbol{\varepsilon}}\Big[w(t)\,\big\|\, s_\theta(\mathbf{x}_t, t) \;-\; \nabla_{\mathbf{x}_t} \log q(\mathbf{x}_t \mid \mathbf{x}_0)\,\big\|^2\Big],$$

with a suitable weight $w(t)$. At sampling time, one integrates the reverse SDE (or the associated probability–flow ODE) from pure noise to obtain data–like samples.

2.3.4 Flow Matching

Flow matching learns a continuous–time transport by training a time–dependent vector field whose flow pushes a simple source distribution to a complex target. It retains the expressivity of neural ODE transports while avoiding many of their training burdens, and has emerged as a strong alternative to diffusion–based samplers (Lipman et al., 2023).

Definition (Flow matching). Given a source p_0 and a target p_1 on $\mathbb{R}^d$, flow matching learns a measurable vector field $v_\theta : \mathbb{R}^d \times [0,1] \to \mathbb{R}^d$ whose induced ODE

$$\frac{d\mathbf{x}_t}{dt} = v_\theta(\mathbf{x}_t, t), \qquad \mathbf{x}_0 \sim p_0,$$

generates a probability path $(p_t)_{t\in[0,1]}$ satisfying the continuity equation

$$\partial_t p_t + \nabla \cdot (p_t\, v_\theta) = 0,$$

with p_1 (approximately) matching the target.

Let $(p_t)_{t\in[0,1]}$ be a chosen bridging path from p_0 to p_1 (e.g., linear interpolation in data space). The ideal vector field $v^\star$ that transports this path solves the continuity equation and, for many choices of p_t, admits a closed-form *conditional* target field. Training then regresses v_θ to this target in mean–squared error.

For the simple linear path $\mathbf{x}_t = (1-t)\mathbf{x}_0 + t\mathbf{x}_1$ with $\mathbf{x}_0 \sim p_0$ and $\mathbf{x}_1 \sim p_1$ independent, the conditional flow matching (CFM) target is

$$u^\star(\mathbf{x}, t \mid \mathbf{x}_1) \;=\; \frac{\mathbf{x}_1 - \mathbf{x}}{1-t} \qquad (t < 1),$$

because $d\mathbf{x}_t/dt = \mathbf{x}_1 - \mathbf{x}_0$ and, conditional on $(\mathbf{x}, t, \mathbf{x}_1)$, one has $\mathbf{x}_0 = \frac{\mathbf{x} - t\mathbf{x}_1}{1-t}$. A practical objective is therefore

$$\mathscr{L}_{\mathrm{CFM}}(\theta) = \mathbb{E}_{t\sim\mathscr{U}(0,1),\, \mathbf{x}_1\sim p_1,\, \mathbf{x}\sim p_t(\cdot\mid\mathbf{x}_1)}\Big[\, \big\| v_\theta(\mathbf{x}, t) - u^\star(\mathbf{x}, t \mid \mathbf{x}_1) \big\|^2 \,\Big],$$

optionally with a time weight $w(t)$ and a small cutoff $t \le 1-\varepsilon$ to avoid the $(1-t)^{-1}$ singularity. This direct regression side–steps reverse–time SDEs and likelihood Jacobians: once trained, one integrates the ODE from $t = 0$ to $t = 1$ (starting at $\mathbf{x}_0 \sim p_0$) to obtain samples from p_1. Alternatives include more sophisticated bridges (e.g., Gaussian couplings) that yield different closed–form targets $u^\star$ but the same regression principle; see (Lipman et al., 2023) for derivations and variants.

2.4 Graph Neural Networks

Designing neural networks that update molecular coordinates imposes symmetry constraints that standard architectures do not natively satisfy. Molecules are naturally represented as graphs (atoms as nodes, bonds as edges) with variable size and rich geometry. In our flow–based coordinate–transport setting we need models that predict new coordinates while respecting the fundamental symmetries of the system. This section sets out the GNN formalism we use and the symmetry requirements for molecular coordinate transformation.

2.4.1 Molecular Representation

Definition (Molecular graph). A molecular system is an undirected, attributed graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ with $|\mathbf{V}| = N$ atoms. Each node $v_i \in \mathbf{V}$ carries a feature vector $\mathbf{h}_i \in \mathbb{R}^{d_v}$ (atom type, charge, etc.) and a coordinate $\mathbf{x}_i \in \mathbb{R}^3$; each edge $(i, j) \in \mathscr{E}$ carries a bond feature $\mathbf{e}_{ij} \in \mathbb{R}^{d_e}$. Let $H = [\mathbf{h}_1, \ldots, \mathbf{h}_N]^\top \in \mathbb{R}^{N \times d_v}$ and $X = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^\top \in \mathbb{R}^{N \times 3}$.

For *coordinate prediction*, the network must be *permutation equivariant* (relabeling atoms only relabels outputs) and *SE(3) equivariant* (rigid motions of the input produce the same rigid motion at the output). By contrast, for *scalar* node or graph properties one requires permutation invariance and SE(3) invariance.

Definition (Permutation equivariance). Let $P \in \{0,1\}^{N \times N}$ be a permutation matrix acting on rows (atom order). A valid coordinate–predictive network f_θ must satisfy

$$f_\theta\big(PH,\, PX\big) \;=\; P\, f_\theta(H, X). \tag{2.13}$$

When edge features are used, they transform as $E \mapsto PEP^\top$; the same equivariance condition applies.

Definition (SE(3) equivariance). For any rotation $R \in \mathrm{SO}(3)$ and translation $\mathbf{t} \in \mathbb{R}^3$, denoting $\mathbf{1}$ the N–vector of ones, we require

$$f_\theta\big(H,\, XR + \mathbf{1}\mathbf{t}^\top\big) \;=\; f_\theta(H, X)\, R + \mathbf{1}\mathbf{t}^\top. \tag{2.14}$$

This guarantees that predicted coordinates transform consistently under rigid motions, as mandated by physical symmetry.

Although a molecule with N atoms lives in a $3N$–dimensional configuration space, chemical connectivity induces strong dependencies (e.g. bond lengths/angles). GNNs exploit these structural priors by operating directly on $\mathbb{G}$, enabling models that satisfy (2.13)–(2.14) while learning effective coordinate updates for our transport maps.

2.4.2 Neural Architectures

Classical graph architectures differ mainly in how they aggregate neighborhood information and whether they encode geometry. For coordinate prediction, the key requirements from §2.4 are permutation equivariance (2.13) and SE(3) equivariance (2.14). Below we summarize the families we build on.

Definition (Message–Passing Neural Networks). At layer ℓ, node states are updated via neighbor messages:

$$\mathbf{m}_i^{(\ell)} = \bigoplus_{j \in \mathcal{N}(i)} M^{(\ell)}\left(\mathbf{h}_i^{(\ell)}, \mathbf{h}_j^{(\ell)}, \mathbf{e}_{ij}, \psi(\mathbf{r}_{ij})\right), \qquad \mathbf{h}_i^{(\ell+1)} = U^{(\ell)}\left(\mathbf{h}_i^{(\ell)}, \mathbf{m}_i^{(\ell)}\right),$$

where $\mathbf{r}_{ij} = \mathbf{x}_j - \mathbf{x}_i$, ψ optionally encodes geometric inputs (e.g., distances), $\bigoplus$ is a permutation–invariant aggregator (sum/mean/max), and $M^{(\ell)}, U^{(\ell)}$ are learned maps.

MPNNs are *permutation equivariant* by construction (the same parameters are shared across nodes and $\bigoplus$ is invariant), but they are not SE(3)–equivariant unless ψ restricts geometry to rotation/translation–invariant quantities (e.g., $\|\mathbf{r}_{ij}\|$). Purely feature–based MPNNs also have a limited receptive field; capturing long–range effects requires depth, which can induce over–smoothing of node states.

Definition (Graph Convolutional Networks). A GCN layer applies a first–order spectral filter:

$$\mathbf{H}^{(\ell+1)} = \sigma\left(\hat{\mathbf{D}}^{-1/2}\hat{\mathbf{A}}\hat{\mathbf{D}}^{-1/2}\mathbf{H}^{(\ell)}\mathbf{W}^{(\ell)}\right),$$

with $\hat{\mathbf{A}} = \mathbf{A} + \mathbf{I}$ and $\hat{\mathbf{D}}$ its degree matrix.

GCNs are a specific MPNN with fixed, degree–normalized weights. They retain permutation equivariance but, like generic MPNNs, do not enforce SE(3) equivariance and inherit locality limitations.

Definition (Graph Attention Networks). GATs use learned attention coefficients to weight neighbors:

$$\alpha_{ij} = \frac{\exp\big(\mathrm{LeakyReLU}\big(\mathbf{a}^\top[\mathbf{W}\mathbf{h}_i \,\|\, \mathbf{W}\mathbf{h}_j]\big)\big)}{\sum_{k\in\mathcal{N}(i)} \exp\big(\mathrm{LeakyReLU}\big(\mathbf{a}^\top[\mathbf{W}\mathbf{h}_i \,\|\, \mathbf{W}\mathbf{h}_k]\big)\big)}, \quad \mathbf{h}_i^{(\ell+1)} = \sigma\Big(\sum_{j\in\mathcal{N}(i)} \alpha_{ij}\,\mathbf{W}\mathbf{h}_j\Big).$$

Attention improves expressivity over fixed weights while preserving permutation equivariance. However, standard GATs still aggregate locally and, unless geometry is encoded invariantly, they are not SE(3)–equivariant.

Definition (Graph Transformers). Self–attention with global receptive field augmented by structural biases:

$$\mathrm{Attn}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top + \mathbf{B}}{\sqrt{d_k}}\right)\mathbf{V},$$

where $\mathbf{B}$ encodes graph structure (e.g., Laplacian/shortest–path/radial bases).

Graph transformers couple global attention with graph priors, enabling long–range interactions crucial for allostery and cooperative motions. Vanilla formulations are permutation equivariant but not SE(3)–equivariant unless geometry enters via invariant encodings only; direct coordinate prediction then becomes inconsistent with (2.14).

Definition (E(3)–equivariant GNNs). Equivariant layers update features and coordinates using only E(3)–invariant scalars and equivariant vector fields. A common form is

$$\mathbf{m}_{ij}^{(\ell)} = \phi_m\big(\mathbf{h}_i^{(\ell)}, \mathbf{h}_j^{(\ell)}, \|\mathbf{r}_{ij}^{(\ell)}\|^2, \mathbf{e}_{ij}\big), \qquad \mathbf{x}_i^{(\ell+1)} = \mathbf{x}_i^{(\ell)} + \sum_{j\neq i} \phi_x\big(\mathbf{m}_{ij}^{(\ell)}\big)\,\big(\mathbf{x}_i^{(\ell)} - \mathbf{x}_j^{(\ell)}\big),$$

$$\mathbf{h}_i^{(\ell+1)} = \phi_h\Big(\mathbf{h}_i^{(\ell)}, \sum_j \mathbf{m}_{ij}^{(\ell)}\Big), \quad \text{with } \mathbf{r}_{ij}^{(\ell)} = \mathbf{x}_j^{(\ell)} - \mathbf{x}_i^{(\ell)}.$$

These constructions guarantee permutation equivariance and SE(3) equivariance by design.

We conclude this geometric section arguing that for predicting coordinates consistently with the symmetries in §2.4, one must use architectures that are permutation equivariant and SE(3) equivariant. E(3)–equivariant GNNs satisfy both and are therefore the appropriate choice for learning coordinate transports.

2.5 Related Work

Having established the theoretical background, we briefly review how the community has addressed the coupled problems of barrier crossing, temperature overlap, and geometric inductive bias in molecular sampling. We focus on neural–enhanced methods for accelerating parallel tempering (PT) via learned transports and on graph neural networks (GNNs) for molecular structure.

A central line of work addresses the *temperature–overlap problem* in PT by *learning* maps that increase overlap between adjacent tempered targets. (Zhang et al., 2025a) demonstrate that neural samplers—normalizing flows and diffusion models—can transport configurations across temperatures with high swap acceptance, thereby reducing the number of required replicas and overall cost. Related work on augmented normalizing flows for molecular dynamics, exemplified by Timewarp (Klein, Foong, et al., 2023), learns expressive invertible transforms from simulation data and offers design ideas relevant to our setting. Most closely related, *Progressive Tempering Sampler with Diffusion (PTSD)* integrates PT with a sequence of diffusion models: a guidance composition of high-temperature models proposes lower-temperature samples, followed by light MCMC refinement, yielding orders-of-magnitude gains in target-evaluation efficiency (Rissanen et al., 2025).

Concurrently, the modeling capacity of normalizing flows has advanced. (Zhai et al., 2024) argue that, with suitable architectures (e.g., transformer–based autoregressive flows), flows can match diffusion–model sample quality while retaining exact likelihoods. At the same time, applying neural samplers directly to unnormalized Boltzmann densities remains delicate: (He et al., 2025) analyze failure modes such as mode collapse and training instabilities. On the theory side, universality guarantees for coupling–based flows (Draxler, Wahl, et al., 2024) and architectural innovations such as Free–form Flows (Draxler, Sorrenson, et al., 2023) broaden the class of invertible transformations available to practitioners. Orthogonally, *Progressive Inference-Time Annealing (PITA)* trains a ladder of diffusion models across temperatures and performs inference-time temperature annealing via a Feynman–Kac/SMC corrector, enabling equilibrium sampling of small N-body and peptide systems in Cartesian coordinates with substantially fewer energy evaluations (Akhound-Sadegh et al., 2025).

Recent efforts emphasize molecular applications, transferability, and adherence to physical constraints. (Tan, Bose, et al., 2025) introduce Sequential Boltzmann Generators that achieve equilibrium sampling of tetra– and hexa–peptides in Cartesian coordinates using transformer–based flows. Transferability across systems without retraining is addressed by (Klein and Noé, 2024), while (Tan, Hassan, et al., 2025) demonstrate large–scale, zero–shot generalization with a high–capacity transferable flow.

Beyond likelihood–based transports, geometry–aware approaches incorporate molecular symmetries and forces. Equivariant flow–matching methods (Klein, Krämer, et al., 2023) preserve SE(3) structure during training, and force–guided bridge matching (Yu et al., 2024) injects physical gradients into the objective. Alternative perspectives learn dynamical operators directly, e.g., Implicit Transfer Operator Learning (Schreiner et al., 2023). Training methodology has likewise progressed: stabilized rectified–flow objectives (Lee et al., 2024) and manifold–aware normalizing flows (Flouris and Konukoglu, 2023) address rugged loss landscapes and constrained configuration spaces.

In summary, the convergence of PT with learned transports and geometric deep learning has opened a practical path toward scalable equilibrium sampling: PT supplies global connectivity; neural transports supply local efficiency; and symmetry–aware representations preserve physical consistency. Open challenges remain in achieving broad transfer across molecular families while maintaining theoretical guarantees and computational efficiency. The next chapter develops our methodology within this emerging landscape.

2.6 Summary

This chapter introduced the background for neural–enhanced PT in molecular equilibrium sampling. We reviewed how the exponential barrier problem limits local MCMC methods, how PT addresses barrier crossing yet suffers from temperature overlap in high dimensions, and how learned transports, particularly normalizing flows, can construct overlap by design. We also outlined the role of GNNs in providing permutation and SE(3)–consistent inductive biases for coordinate prediction. These ingredients motivate the methodological development that follows: learning bijective, geometry–aware transports to improve swap acceptance while preserving exact sampling guarantees.

Chapter 3

Methodology

Everything should be made as simple as possible,
but not simpler.
— Albert Einstein (1879–1955)

As established in the previous chapter, the core difficulty in PT is the exponential decay of adjacent-replica swap acceptance as molecular complexity grows. Classical fixes tune the temperature ladder (e.g., reducing gaps), but this drives the number of replicas to grow exponentially (Predescu et al., 2004). Instead, neural transport maps learn transformations that construct overlap between temperature distributions, rather than relying on incidental overlap. This chapter develops such transports and details our methodology for PT acceleration.

3.1 Problem Formulation

Let $(\Omega, \mathscr{F}, \lambda)$ be the molecular configuration space with Lebesgue measure λ, where $\Omega = \mathbb{R}^{3N}$ are Cartesian coordinates of an N-atom system (or a smooth constraint manifold). All target measures are assumed absolutely continuous w.r.t. λ, hence admit densities $p = \mathrm{d}P/\mathrm{d}\lambda$. In particular, the tempered Boltzmann density and corresponding measure are

$$p_\beta(x) \;=\; Z_\beta^{-1} e^{-\beta U(x)}, \qquad P_\beta(\mathrm{d}x) \;=\; p_\beta(x)\,\lambda(\mathrm{d}x), \qquad \beta > 0, \tag{3.1}$$

with partition function Z_β and potential energy $U(x)$.

Parallel tempering (PT) maintains M replicas with joint state in $(\Omega^M, \mathscr{F}^{\otimes M})$. Each replica evolves via a local Markov kernel that preserves its tempered Boltzmann measure; PT alternates these local steps with *swap moves* between replica pairs (see Subsection 2.2.3).

After every L single-replica steps, a swap kernel acts on a chosen pair: given $(x,y) \in \Omega \times \Omega$, propose (y,x) and accept/reject by Metropolis–Hastings.

To analyze a specific adjacent pair $(k,k+1)$, we work on $\Omega \times \Omega$ with product target

$$\Pi := P_{\beta_k} \otimes P_{\beta_{k+1}}, \qquad \pi(x,y) = p_{\beta_k}(x)\, p_{\beta_{k+1}}(y) \ \text{(density w.r.t. } \lambda \otimes \lambda).$$

For current $(x,y) \sim \Pi$ and proposal kernel $Q\big((x,y), \mathrm{d}(x',y')\big)$, the MH acceptance is

$$\alpha\big((x,y),(x',y')\big) = \min\left\{1,\ \frac{p_{\beta_{k+1}}(x')\, p_{\beta_k}(y')\, Q\big((x',y'), \mathrm{d}(x,y)\big)}{p_{\beta_k}(x)\, p_{\beta_{k+1}}(y)\, Q\big((x,y), \mathrm{d}(x',y')\big)}\right\}. \tag{3.2}$$

See (Douc and Cor, 2015) for a concise derivation on product spaces and detailed balance.

For the *naive coordinate swap* $g(x,y) = (y,x)$, the proposal is deterministic, so $Q\big((x,y), \mathrm{d}(x',y')\big) = \delta_{(y,x)}\big(\mathrm{d}(x',y')\big)$ and the Hastings factor cancels. With $(x',y') = (y,x)$ we obtain

$$\alpha_{\text{swap}}(x,y) = \min\left\{1,\ \frac{p_{\beta_k}(y)\, p_{\beta_{k+1}}(x)}{p_{\beta_k}(x)\, p_{\beta_{k+1}}(y)}\right\}.$$

Specializing to Boltzmann densities (3.1), normalizing constants cancel and

$$\frac{p_{\beta_k}(y)\, p_{\beta_{k+1}}(x)}{p_{\beta_k}(x)\, p_{\beta_{k+1}}(y)} = \exp\Big[(\beta_k - \beta_{k+1})\big(U(y) - U(x)\big)\Big],$$

hence the *naive swap acceptance* is

$$\boxed{\alpha_{\text{naive}}(x,y) = \min\Big\{1,\ \exp\Big[(\beta_k - \beta_{k+1})\big(U(y) - U(x)\big)\Big]\Big\}} \tag{3.3}$$

Temperature-Overlap Problem. For high-dimensional molecular systems,

$$\mathbb{E}_{(x,y)\sim P_{\beta_k} \otimes P_{\beta_{k+1}}}\big[\alpha_{\text{naive}}(x,y)\big] \to 0$$

as system complexity increases, because configurations typical at one temperature have negligible probability under the other, yielding vanishing overlap.

To overcome this, we replace *naive* swaps by a *learned transport swap*. Let $T_\theta : \Omega \to \Omega$ be an invertible, measurable map with inverse T_θ^{-1}. Define the deterministic proposal

$$g(x,y) = \big(T_\theta^{-1}(y),\ T_\theta(x)\big), \tag{3.4}$$

which is an *involution* since $T_\theta^{-1}(T_\theta(x)) = x$ and $T_\theta(T_\theta^{-1}(y)) = y$, hence $g \circ g = \mathrm{id}$. A plain deterministic proposal $Q\big((x,y),\cdot\big) = \delta_{g(x,y)}(\cdot)$ requires mutual support to use the standard MH ratio (3.2); for a Dirac proposal this holds exactly when $g \circ g = \mathrm{id}$. In the deterministic case, the acceptance takes the form

$$\alpha(x,y) \;=\; \min\left\{1,\; \frac{\pi\big(g(x,y)\big)\,\big|\det J_g(x,y)\big|}{\pi(x,y)}\right\}, \tag{3.5}$$

whose derivation is given in Appendix A.3.

Plugging (3.4) into (3.5) and using change of variables yields the *flow-enhanced acceptance*

$$\alpha_{\text{flow}}(x,y) \;=\; \min\big\{1,\, \exp\big[\Delta_{\text{flow}}(x,y)\big]\big\}, \tag{3.6}$$

with

$$\begin{aligned}\Delta_{\text{flow}}(x,y) = &-\beta_k U\big(T_\theta^{-1}(y)\big) - \beta_{k+1} U\big(T_\theta(x)\big) + \beta_k U(x) + \beta_{k+1} U(y) \\ &+ \log\big|\det J_{T_\theta}(x)\big| + \log\big|\det J_{T_\theta^{-1}}(y)\big|.\end{aligned} \tag{3.7}$$

The Jacobian terms are the Radon–Nikodym derivatives of the pushforwards $T_\theta \sharp P_{\beta_k}$ and $T_\theta^{-1} \sharp P_{\beta_{k+1}}$ (see Appendix A.4 for a derivation).

Finally, we state the exactness guarantee. In Appendix A.1 we prove detailed balance for MH with a stochastic proposal under mutual support. The following lemma gives the deterministic counterpart.

Lemma 3.1.1 (MH with an involutive deterministic proposal)**.** *Let $g : \Omega \times \Omega \to \Omega \times \Omega$ be a C^1 involution ($g \circ g = \mathrm{id}$). Then the MH kernel with deterministic proposal $(x,y) \mapsto g(x,y)$ and acceptance* (3.5) *satisfies detailed balance w.r.t. the product target density $\pi(x,y) = p_{\beta_k}(x) p_{\beta_{k+1}}(y)$.*

Corollary 3.1.2 (Flow–swap preserves $P_{\beta_k} \otimes P_{\beta_{k+1}}$)**.** *Let $T_\theta : \Omega \to \Omega$ be a C^1 diffeomorphism with inverse T_θ^{-1}, and define $g(x,y) = (T_\theta^{-1}(y), T_\theta(x))$. Then g is a C^1 involution and, by Lemma 3.1.1, the MH kernel satisfies detailed balance for $\Pi = P_{\beta_k} \otimes P_{\beta_{k+1}}$ with acceptance given by* (3.6)*.*

Thus, flow-enhanced PT remains exact while potentially achieving substantially higher swap acceptance than naive exchanges. We now develop neural architectures for learning effective transports T_θ.

3.2 Architectures

We propose three architectures that trade off molecular inductive bias and expressive power: (1) *coordinate-only* flows that treat molecules as flat vectors; (2) *graph-based* flows that encode local chemical connectivity and partial symmetries; and (3) *transformer* flows that drop geometric constraints to capture global interactions via attention.

All models are normalizing flows adapted to PT. The coordinate-only design follows RealNVP (Dinh et al., 2017). The graph and transformer designs borrow ideas from Timewarp (Klein, Foong, et al., 2023) but target a different objective: we learn *potential–energy* transports between temperatures for PT swaps, rather than *temporal* evolution in MD.

3.2.1 PTSwapFlow

We start with a minimal baseline trained on a single-peptide dataset and with no explicit chemical structure. It treats each molecule as an unlabeled point cloud, and is therefore peptide-specific: it does not generalize to unseen systems.

Let $\mathbf{x} \in \mathbb{R}^{3N}$ be flattened Cartesian coordinates. PTSwapFlow stacks RealNVP coupling layers with alternating *atomic* masks. A binary mask $\mathbf{m} \in \{0,1\}^{3N}$ groups full atomic triplets:

$$m_{3j+k} = j \bmod 2, \quad j \in \{0,\dots,N-1\}, \quad k \in \{0,1,2\} \tag{3.8}$$

This groups complete atomic coordinates (all three spatial dimensions) together, alternating between even and odd atoms across coupling layers. The coupling transformation computes scale and shift parameters via multilayer perceptron:

$$\mathbf{s}, \mathbf{t} = \mathrm{MLP}(\mathbf{x} \odot (1 - \mathbf{m}))$$

and applies the affine transformation:

$$\mathbf{y} = \mathbf{x} \odot (1 - \mathbf{m}) + (\mathbf{x} \odot \mathbf{m}) \odot \exp(\mathbf{s}) + \mathbf{t} \odot \mathbf{m} \tag{3.9}$$

The exponential ensures positive scaling factors, guaranteeing invertibility. The affine form provides computational tractability while maintaining sufficient expressivity for molecular coordinate transformations. The inverse transformation recovers the original coordinates analytically:

$$\mathbf{x} = \mathbf{y} \odot (1 - \mathbf{m}) + \frac{(\mathbf{y} \odot \mathbf{m}) - \mathbf{t}}{\exp(\mathbf{s})} \odot \mathbf{m} \tag{3.10}$$

Note that exact invertibility follows because the conditioning subset $\mathbf{x} \odot (1 - \mathbf{m})$ passes through unchanged, eliminating any information loss. The Jacobian matrix $J = \frac{\partial \mathbf{y}}{\partial \mathbf{x}}$ has special structure due to the coupling design:

$$J_{ij} = \begin{cases} 1 & \text{if } i = j \text{ and } m_i = 0 \text{ (unchanged coordinates)} \\ \exp(s_i) & \text{if } i = j \text{ and } m_i = 1 \text{ (transformed coordinates)} \\ \frac{\partial s_i}{\partial x_j} x_i \exp(s_i) + \frac{\partial t_i}{\partial x_j} & \text{if } i \neq j, m_i = 1, m_j = 0 \text{ (coupling)} \\ 0 & \text{otherwise} \end{cases}$$

Since s_i and t_i depend only on unchanged coordinates ($m_j = 0$), the matrix has block structure with determinant $\prod_{i:m_i=1} \exp(s_i)$, yielding:

$$\log |\det J| = \sum_{i:m_i=1} s_i \tag{3.11}$$

This closed-form log-determinant enables exact likelihoods and plugs directly into the swap acceptance in Equation (3.7). A worked 2D example is given in Appendix A.2.

However, this coordinate-only approach lacks chemical awareness. The natural progression was therefore to enhance our structure-agnostic flow with molecular information. Our ultimate objective was to develop *transferable* flows that can generalize across unseen molecular systems. The most straightforward extension would involve augmenting atomic coordinates within the RealNVP-style coupling layers with learnable chemical element embeddings (C, H, N, O). However, this naive concatenation strategy appeared fundamentally flawed. Simply appending atom type embeddings to coordinates would still treat molecules as unstructured collections of labeled points, completely ignoring the connectivity patterns that define molecular behavior and properties. Recognizing these limitations, we opted for a more sophisticated framework as next that explicitly incorporates graph-based molecular connectivity as our primary approach to encoding chemical information.

3.2.2 PTSwapGraphFlow

Rather than treating molecules as unstructured coordinate vectors, the graph-aware architecture models each conformation as a *radius graph* built from 3D coordinates. Nodes are atoms with types $\tau_i \in \{\mathrm{H}, \mathrm{C}, \mathrm{N}, \mathrm{O}\}$, and edges connect spatial neighbors within a distance threshold τ. In this setting, the transport T_θ is parameterized by symmetry-aware coupling layers that use local message passing to compute per-atom scale and shift. Concretely, we map $\mathbf{x}^{(T_{\text{low}})} \mapsto \mathbf{x}^{(T_{\text{high}})} = T_\theta(\mathbf{x}^{(T_{\text{low}})}, \boldsymbol{\tau})$, extending the coordinate-only design by conditioning

on atom types $\boldsymbol{\tau} = \{\tau_1, \ldots, \tau_N\}$. While a fully connected graph would maximize expressivity, we adopt a high-bias, geometry-respecting choice, sparse, distance-based neighborhoods, for stability and efficiency, leaving the low-bias alternative to the transformer in the next subsection.

The key enhancement is to build node features and aggregations from *invariant* quantities wherever possible, preserving molecular symmetries by construction. Atom i carries a learned type embedding $\mathbf{h}_i = W_{\text{embed}}[\mathbf{e}_{\tau_i}] \in \mathbb{R}^{d_h}$, and its neighborhood is

$$\mathcal{N}_i = \{\, j \neq i : \ \|\mathbf{x}_j - \mathbf{x}_i\|_2 < \tau \text{ nm} \,\}. \tag{3.12}$$

These neighborhoods define the edges of the radius graph and supply local chemical context.

Each coupling layer retains the affine update from Equations (3.9)–(3.10), but the scale and shift are now produced by graph-based message passing. We summarize the neighborhood by an invariant scalar and a vector aggregate:

$$\bar{d}_i^{(\ell)} = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \left\|\mathbf{x}_j^{(\ell)} - \mathbf{x}_i^{(\ell)}\right\|_2, \tag{3.13}$$

$$s_i^{(\ell)} = \text{MLP}_s^{(\ell)}\Big([\mathbf{h}_i,\ \bar{d}_i^{(\ell)}]\Big), \tag{3.14}$$

$$\mathbf{t}_i^{(\ell)} = \frac{\text{MLP}_t^{(\ell)}(\mathbf{h}_i)}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \big(\mathbf{x}_j^{(\ell)} - \mathbf{x}_i^{(\ell)}\big). \tag{3.15}$$

Here, the same small networks $\text{MLP}_s^{(\ell)}$ and $\text{MLP}_t^{(\ell)}$ are shared across nodes, and the sums over $\mathcal{N}_i$ implement permutation-invariant message aggregation.

This asymmetric construction reflects the different geometric roles of scaling and translation. The scale $s_i^{(\ell)}$ depends only on *scalar* neighborhood statistics and type, making it rotationally and translationally invariant while adapting to local density (small scales in crowded regions to avoid clashes, larger scales in sparse regions). In contrast, the shift $\mathbf{t}_i^{(\ell)}$ is a type-weighted average of *relative* displacement vectors, preserving full directional information and yielding an $E(3)$-equivariant update:

$$\mathbf{t}_i^{(\ell)}(R\mathbf{x} + c) = R\,\mathbf{t}_i^{(\ell)}(\mathbf{x}), \qquad \forall R \in SO(3),\ c \in \mathbb{R}^3.$$

The coupling then applies the per-atom isotropic affine map on the masked subset and leaves the complement unchanged, maintaining exact invertibility as before.

To batch molecules with different sizes, we pad to the maximum number of atoms N in the batch using dummy atoms (type -1, zero coordinates) and apply masks so that message passing and Jacobian computations ignore padded positions.

3.2.3 PTSwapTransformerFlow

Whereas graph-based flows emphasize local chemistry, the transformer variant deliberately relaxes geometric constraints and models *global* interactions via self–attention (Vaswani et al., 2023). This is a conscious trade-off: we give up explicit $E(3)$-equivariance in exchange for higher expressivity, using fully connected attention to capture long-range correlations that empirically improve swap acceptance.

Following the augmented-flow idea of Timewarp (Klein, Foong, et al., 2023), we extend the state with per-atom auxiliary variables. Instead of modeling only $\mu_\beta(\mathbf{x})$, the model targets the joint

$$\mu_{\text{aug}}(\mathbf{x},\mathbf{v}) \propto \exp\Big(-U(\mathbf{x})/(k_B T)\Big)\,\mathcal{N}(\mathbf{v};\mathbf{0},\mathbf{I}), \tag{3.16}$$

where $\mathbf{v}_i \sim \mathcal{N}(\mathbf{0},\mathbf{I})$ are nonphysical latents. These latents carry no chemistry; they enlarge the transformation space and, coupled symmetrically across atoms, preserve permutation equivariance.

The flow stacks L coupling layers that *alternate* the transformed variable type: layer ℓ updates all coordinates if ℓ is even, and all auxiliaries if ℓ is odd. This schedule lets each variable type condition the other while maintaining exact invertibility. Padding for variable-sized molecules again uses dummy atoms (type -1, zero coordinates/latents) together with a fixed attention mask $\mathbf{M} \in \{0,1\}^N$ across all layers; masked positions receive $-\infty$ logits before softmax, so they never attend or get attended to.

At layer ℓ, we build per-atom features by concatenation

$$\mathbf{f}_i^{(\ell)} = \big[\,\mathbf{h}_i,\, \mathbf{x}_i^{(\text{init})},\, \mathbf{v}_i^{(\text{init})},\, \mathbf{z}_{\text{cond},i}^{(\ell)},\, \text{RFF}\big(\mathbf{x}_i^{(\text{init})}\big)\,\big], \tag{3.17}$$

where $\mathbf{h}_i \in \mathbb{R}^{d_h}$ is the atom type embedding, $\mathbf{z}_{\text{cond},i}^{(\ell)} = \mathbf{v}_i^{(\ell)}$ when updating coordinates (even ℓ), and $\mathbf{z}_{\text{cond},i}^{(\ell)} = \mathbf{x}_i^{(\ell)}$ when updating auxiliaries (odd ℓ). Random Fourier features term $\text{RFF}(\mathbf{x}_i^{(\text{init})})$ provides a fixed random sinusoidal positional encoding that supplies smooth, distance-aware cues to attention.

We implement random Fourier features (RFF) by sampling frequencies $\boldsymbol{\omega}_r \sim \mathcal{N}(\mathbf{0}, \ell^{-2}\mathbf{I})$ and phases $b_r \sim \mathrm{Unif}[0, 2\pi]$ once at initialization, and defining

$$\mathrm{RFF}(\mathbf{x}) = \sqrt{\tfrac{2}{m}} \begin{bmatrix} \cos(\boldsymbol{\Omega}\mathbf{x} + \mathbf{b}) \\ \sin(\boldsymbol{\Omega}\mathbf{x} + \mathbf{b}) \end{bmatrix} \in \mathbb{R}^{d_{\mathrm{RFF}}}, \qquad m = \tfrac{1}{2} d_{\mathrm{RFF}}, \quad \boldsymbol{\Omega} = \begin{bmatrix} \boldsymbol{\omega}_1^\top \\ \vdots \\ \boldsymbol{\omega}_m^\top \end{bmatrix}, \ \mathbf{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}. \tag{3.18}$$

The inner products of these features approximate an RBF kernel, thereby furnishing rich positional signals without hard-coding geometry (Rahimi and Recht, 2007).

Each coupling layer computes flow parameters in three steps. First, an input projection maps features to the model width:

$$\mathrm{MLP}_{\mathrm{in}} : \mathbb{R}^{N \times d_{\mathrm{feat}}} \to \mathbb{R}^{N \times d_{\mathrm{model}}}, \qquad \mathbf{F}^{(\ell)} = \mathrm{MLP}_{\mathrm{in}}\big(\mathbf{f}_1^{(\ell)}, \ldots, \mathbf{f}_N^{(\ell)}\big),$$

Second, multi-head self–attention captures global dependencies under the padding mask

$$\tilde{\mathbf{F}}^{(\ell)} = \mathrm{MHA}\big(\mathbf{F}^{(\ell)}, \mathbf{F}^{(\ell)}, \mathbf{F}^{(\ell)}; \mathbf{M}\big).$$

Third, separate heads predict per-atom, per-axis scale and shift

$$\mathbf{s}^{(\ell)} = \mathrm{MLP}^{s}_{\mathrm{out}}\big(\tilde{\mathbf{F}}^{(\ell)}\big) \in \mathbb{R}^{N \times 3},$$
$$\mathbf{t}^{(\ell)} = \mathrm{MLP}^{t}_{\mathrm{out}}\big(\tilde{\mathbf{F}}^{(\ell)}\big) \in \mathbb{R}^{N \times 3}.$$

The coupling transformation updates the target variables $\mathbf{z}^{(\ell)}$ while leaving conditioning variables unchanged:

$$\mathbf{z}_i^{(\ell+1)} = \mathbf{z}_i^{(\ell)} \odot \exp(\mathbf{s}_i^{(\ell)}) + \mathbf{t}_i^{(\ell)} \tag{3.19}$$

with analytical inverse similar to Equation (3.10). The log-determinant computation follows the same principle as Equation (3.11), summing the scale parameters across all coupling layers for both coordinate and auxiliary variable transformations. The attention mechanism enables each atom's transformation to depend on all other atoms simultaneously, capturing global molecular correlations without distance constraints while maintaining exact invertibility through the coupling structure.

3.3 Molecular Symmetries

Our three architectures offer different symmetry guarantees. *PTSwapFlow* has no built–in geometric symmetry and relies on data augmentation. *PTSwapTransformerFlow* is permutation–equivariant (via shared per-atom processing and attention with a padding mask) but is not $E(3)$–equivariant; any rotational/translation behavior is learned from data. *PTSwapGraphFlow* incorporates symmetry-aware components and enjoys partial theoretical guarantees.

For the graph architecture, consider the $E(3)$ action $\mathbf{x} \mapsto R\mathbf{x} + c$ with $R \in SO(3)$ and $c \in \mathbb{R}^3$. The coupling layer (Equation (3.9)) uses per-atom *scales* s_i and *shifts* $\mathbf{t}_i$ designed to respect these symmetries where possible. From Equation (3.14), the scales depend only on invariant quantities,

$$s_i = \exp\Big(\mathrm{MLP}_s\big([\mathbf{h}_i,\, \bar{d}_i]\big)\Big), \qquad \bar{d}_i = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \|\mathbf{x}_j - \mathbf{x}_i\|_2,$$

so $s_i(R\mathbf{x} + c) = s_i(\mathbf{x})$ because $\mathbf{h}_i$ is coordinate–independent and pairwise distances are invariant. From Equation (3.15), the shifts are scalar–weighted sums of relative vectors,

$$\mathbf{t}_i = \frac{c_i}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} (\mathbf{x}_j - \mathbf{x}_i), \qquad c_i = \mathrm{MLP}_t(\mathbf{h}_i),$$

hence $\mathbf{t}_i(R\mathbf{x} + c) = R\,\mathbf{t}_i(\mathbf{x})$ (equivariance) because c_i is invariant and $(\mathbf{x}_j - \mathbf{x}_i)$ rotates.

Combining these in the per-atom affine update $\mathbf{y}_i = \exp(s_i)\mathbf{x}_i + \mathbf{t}_i$ yields *rotation equivariance*

$$\mathbf{y}_i(R\mathbf{x}) = \exp(s_i)R\mathbf{x}_i + R\mathbf{t}_i = R\big(\exp(s_i)\mathbf{x}_i + \mathbf{t}_i\big) = R\,\mathbf{y}_i(\mathbf{x}).$$

Exact translation equivariance, however, fails in general

$$\mathbf{y}_i(\mathbf{x} + c) = \exp(s_i)(\mathbf{x}_i + c) + \mathbf{t}_i = \mathbf{y}_i(\mathbf{x}) + \exp(s_i)c,$$

which equals $\mathbf{y}_i(\mathbf{x}) + c$ only if $s_i = 0$. A standard remedy is the centroid reparameterization $\mathbf{y}_i = \mu + \exp(s_i)(\mathbf{x}_i - \mu) + \mathbf{t}_i$, with $\mu = \frac{1}{N}\sum_k \mathbf{x}_k$, which restores *exact* translation equivariance while preserving invertibility; we show empirically in Section 4.4, this variant yields translation errors at numerical tolerance. Permutation equivariance is achieved by constructing radius graphs from coordinates (not indices) and using sum aggregations; batching with variable N is handled via padding masks so that padded nodes neither send nor receive messages. Under these choices, the graph parameterization is permutation–equivariant, rotation–equivariant, and approximately translation–equivariant (or exactly so under the centroid form). We empirically validate these claims in Section 4.4.

3.4 Learning Objective

Training the flow for PT amounts to learning a bijection T_θ that *constructs overlap* between adjacent temperatures so that swap proposals are readily accepted. Formally, let $p_{\beta_k}, p_{\beta_{k+1}}$ be the Boltzmann densities at β_k and β_{k+1}. The transport desideratum is as follows

$$x \sim P_{\beta_k} \;\; T_\theta(x) \sim P_{\beta_{k+1}}, \qquad y \sim P_{\beta_{k+1}} \;\; T_\theta^{-1}(y) \sim P_{\beta_k}.$$

Equivalently, the pushforwards $T_\theta \sharp P_{\beta_k}$ and $T_\theta^{-1} \sharp P_{\beta_{k+1}}$ should match the opposite targets. A natural way to encode this is the *bidirectional* KL divergence (Kullback and Leibler, 1951)

$$\mathscr{L}_{\mathrm{KL}}(\theta) = D_{\mathrm{KL}}\big(P_{\beta_{k+1}} \parallel T_\theta \sharp P_{\beta_k}\big) + D_{\mathrm{KL}}\big(P_{\beta_k} \parallel T_\theta^{-1} \sharp P_{\beta_{k+1}}\big),$$

which vanishes iff T_θ is an exact transport between the two. Change of variables gives the standard identity that minimizing the bidirectional KL divergence $\mathscr{L}_{\mathrm{KL}}(\theta)$ is equivalent to minimizing the negative log-likelihood (NLL) of each temperature's samples evaluated under the flow-transformed distribution of the other (see A.6)

$$\begin{aligned}\mathscr{L}_{\mathrm{NLL}}(\theta) = &-\mathbb{E}_{y\sim p_{\beta_{k+1}}}\Big[\log\big(p_{\beta_k}(T_\theta^{-1}(y))\,|\det J_{T_\theta^{-1}}(y)|\big)\Big] \\ &-\mathbb{E}_{x\sim p_{\beta_k}}\Big[\log\big(p_{\beta_{k+1}}(T_\theta(x))\,|\det J_{T_\theta}(x)|\big)\Big]. \end{aligned} \tag{3.20}$$

Using $-\log p_\beta(z) = \beta U(z) + \log Z_\beta$, and discarding θ-independent constants ($\log Z_\beta$), we obtain the per-temperature NLL terms

$$\begin{aligned}\mathscr{L}_{\mathrm{NLL}}(\theta) = &\;\mathbb{E}_{y\sim p_{\beta_{k+1}}}\Big[\beta_k U\big(T_\theta^{-1}(y)\big) - \log|\det J_{T_\theta^{-1}}(y)|\Big] \\ &+\mathbb{E}_{x\sim p_{\beta_k}}\Big[\beta_{k+1} U\big(T_\theta(x)\big) - \log|\det J_{T_\theta}(x)|\Big]. \end{aligned} \tag{3.21}$$

Intuitively, each bracket rewards the flow for mapping samples from one temperature into high-density regions of the other (small βU), while penalising unjustified local expansion via the Jacobian. This is precisely the mechanism by which the flow *creates* overlap between marginals. It is noticeable that this expression is quite similar to the flow-enhanced acceptance criterion Δ_{flow} introduced in Equation (3.7). In particular, a direct rearrangement yields the

per-pair identity

$$\underbrace{\begin{aligned}&\beta_{k+1}U\big(T_\theta(x)\big) - \log\big|\det J_{T_\theta}(x)\big| \\ &\quad + \beta_k U\big(T_\theta^{-1}(y)\big) - \log\big|\det J_{T_\theta^{-1}}(y)\big|\end{aligned}}_{L_{\text{NLL, pair}}(x,y;\theta)} = -\Delta_{\text{flow}}(x,y) + \beta_k U(x) + \beta_{k+1}U(y) + \text{const.} \tag{3.22}$$

Hence, for *fixed* (x,y),

$$\nabla_\theta L_{\text{NLL,pair}}(x,y;\theta) = \nabla_\theta\big[-\Delta_{\text{flow}}(x,y)\big].$$

This formalises the statement that NLL minimisation *pushes in the same direction* as increasing per-pair acceptance; the extra input-energy sum $\beta_k U(x) + \beta_{k+1}U(y)$ is a baseline independent of θ that sets how hard it is for a particular pair to satisfy $\Delta_{\text{flow}} \geq 0$.

However, as introduced in Section 3.2, the *PTSwapTransformerFlow* operates on *augmented* states. For the adjacent pair we write

$$z_{\text{low}} = (x, v_x), \qquad z_{\text{high}} = (y, v_y),$$

where $x, y \in \Omega$ are positions and $v_x, v_y \in \mathbb{R}^{3N}$ are toy (non-physical) velocities drawn from the temperature-independent standard Gaussian. The target at inverse temperature β factorises then as

$$p_\beta(x,v) \;\propto\; \exp\big(-\beta\, U(x) - \tfrac{1}{2}\|v\|^2\big).$$

Let the forward map send $z_{\text{low}} \mapsto z'_{\text{high}} = (x', v'_x) := T_\theta(x, v_x)$ (low→high), and the inverse map send $z_{\text{high}} \mapsto z'_{\text{low}} = (y', v'_y) := T_\theta^{-1}(y, v_y)$ (high→low). Replacing the NLL brackets by their augmented versions yields

$$\begin{aligned}L_{\text{NLL,aug}}(z_{\text{low}}, z_{\text{high}}; \theta) = &\Big[\beta_{k+1}\, U(x') + \tfrac{1}{2}\|v'_x\|^2 - \log\big|\det J_{T_\theta}(x, v_x)\big|\Big] \\ &+ \Big[\beta_k\, U(y') + \tfrac{1}{2}\|v'_y\|^2 - \log\big|\det J_{T_\theta^{-1}}(y, v_y)\big|\Big],\end{aligned} \tag{3.23}$$

where $J_{T_\theta}(x, v_x)$ denotes the full Jacobian of the augmented map $T_\theta : \mathbb{R}^{6N} \to \mathbb{R}^{6N}$ with respect to the concatenated variables (x, v_x) (analogously for T_θ^{-1}).

The corresponding *augmented* acceptance exponent, consistent with Eq. (3.7), is

$$\begin{aligned}\Delta_{\text{aug}}(z_{\text{low}}, z_{\text{high}}) = &-\beta_k U(y') - \beta_{k+1} U(x') + \beta_k U(x) + \beta_{k+1} U(y) \\ &- \tfrac{1}{2}\|v_y'\|^2 - \tfrac{1}{2}\|v_x'\|^2 + \tfrac{1}{2}\|v_x\|^2 + \tfrac{1}{2}\|v_y\|^2 \\ &+ \log\left|\det J_{T_\theta}(x, v_x)\right| + \log\left|\det J_{T_\theta^{-1}}(y, v_y)\right|. \end{aligned} \tag{3.24}$$

Exactly as in the positions-only identity, we then have

$$L_{\text{NLL,aug}}(z_{\text{low}}, z_{\text{high}}; \theta) = -\Delta_{\text{aug}}(z_{\text{low}}, z_{\text{high}}) + \beta_k U(x) + \beta_{k+1} U(y) + \tfrac{1}{2}\|v_x\|^2 + \tfrac{1}{2}\|v_y\|^2 + \text{const}, \tag{3.25}$$

so for a fixed pair $(z_{\text{low}}, z_{\text{high}})$, $\nabla_\theta L_{\text{NLL,aug}} = \nabla_\theta\big(-\Delta_{\text{aug}}\big)$. In practice, we made an error, as we forgot to including the kinetic terms which ended up being important: without them, the model could artificially inflate $\log|\det J|$ by rescaling the velocity channels while barely changing positions. The $\frac{1}{2}\|v\|^2$ contributions "charge" such manoeuvres, aligning the augmented objective with the intended target. Luckily, we realized of this bug and solved this issue.

3.5 Training

Training flow-enhanced PT requires learning invertible transformations between adjacent temperature distributions while maintaining numerical stability across diverse molecular systems. We train one flow per adjacent temperature pair (β_k, β_{k+1}), yielding $M-1$ independent models for M replicas. The training pipeline combines physics-informed likelihood objectives with robust data processing strategies to handle variable molecular sizes, temporal correlations, and energy outliers that arise in molecular simulations.

3.5.1 Optimization

For a given temperature pair, the empirical objective is the Monte Carlo estimate

$$\begin{aligned}\widehat{\mathscr{L}}_{\text{ACC}}(\theta) = &\frac{1}{m}\sum_{i=1}^{m}\Big[\beta_{\text{low}} U\big(T_\theta^{-1}(x_{\text{high}}^{(i)})\big) - \log\big|\det J_{T_\theta^{-1}}(x_{\text{high}}^{(i)})\big|\Big] \\ &+ \frac{1}{n}\sum_{j=1}^{n}\Big[\beta_{\text{high}} U\big(T_\theta(x_{\text{low}}^{(j)})\big) - \log\big|\det J_{T_\theta}(x_{\text{low}}^{(j)})\big|\Big], \end{aligned} \tag{3.26}$$

with minibatches $\{x_{\text{high}}^{(i)}\}_{i=1}^m \sim p_{\text{high}}$ and $\{x_{\text{low}}^{(j)}\}_{j=1}^n \sim p_{\text{low}}$. Training proceeds by optimizing this objective using the Adam optimizer (Kingma and Ba, 2017), chosen for its adaptive

learning rate properties. Adam algorithm maintains exponentially decaying averages of past gradients and their squared values to compute per-parameter adaptive learning rates. Given $g_t = \nabla_\theta \widehat{\mathscr{L}}^{(t)}_{\mathrm{ACC}}(\theta)\big|_{\theta=\theta_t}$, the moment estimates update as

$$m_t = \rho_1 m_{t-1} + (1-\rho_1) g_t$$
$$v_t = \rho_2 v_{t-1} + (1-\rho_2) g_t^2$$

with bias-corrected forms $\hat{m}_t = m_t/(1-\rho_1^t)$ and $\hat{v}_t = v_t/(1-\rho_2^t)$. The parameter update rule becomes:

$$\theta_{t+1} = \theta_t - \alpha \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \varepsilon} \tag{3.27}$$

where α is the (base) learning rate, $\rho_1, \rho_2 \in (0,1)$ are the EMA decay coefficients for the first and second moment estimates, and $\varepsilon > 0$ is a small numerical stabilizer. Training stability is ensured through gradient clipping and ReduceLROnPlateau scheduling monitoring validation loss. Early stopping prevents overfitting.

Energy gating prevents gradient instability by filtering samples exceeding physically reasonable energy thresholds during loss computation. Our implementation employs a two-stage energy regularization scheme. First, energies are hard-clamped at $E_{\max}$ kJ/mol to prevent numerical overflow:

$$U_{\text{clamped}} = \min(U, E_{\max})$$

Second, a soft regularization applies logarithmic compression to energies exceeding E_{cut} kJ/mol:

$$U_{\text{reg}} = \begin{cases} U & \text{if } U \le E_{\text{cut}} \\ E_{\text{cut}} + \log(1 + U - E_{\text{cut}}) & \text{if } U > E_{\text{cut}} \end{cases}$$

This regularization maintains gradient flow while preventing extreme energies from dominating the loss landscape. When all samples in a batch exceed the energy threshold, a sentinel value is returned, effectively skipping the batch and avoiding gradient updates that could destabilize training. The implementation employs peptide-specific target caching to avoid repeated OpenMM system initialization, which otherwise triggers parser errors during multi-peptide training. Our framework supports both single-peptide training and multi-peptide training modes.

3.5.2 Data Pipeline

The training pipeline incorporates several preprocessing steps to enhance data quality and model robustness. Subsampling reduces computational overhead by extracting every N-th

frame from PT trajectories, *and* it thins temporally correlated sequences so consecutive training examples are more weakly related, yielding a higher effective sample size and more stable gradient estimates. Coordinate centering removes translational degrees of freedom by subtracting the molecular centroid: $\mathbf{x}_{\text{centered}} = \mathbf{x} - \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i$, simplifying the learning task. Chirality filtering preserves the L-amino acid configuration by detecting and removing conformations with incorrect stereochemistry based on dihedral angle constraints, preventing the model from learning unphysical molecular geometries.

Most critically, random rotation augmentation enhances rotational invariance by applying consistent 3D rotations to coordinate pairs. For each training sample pair $(\mathbf{x}_{\text{low}}, \mathbf{x}_{\text{high}})$, we generate a random rotation matrix using Rodrigues' formula. Given a random unit axis $\mathbf{v} \sim \mathcal{N}(0,\mathbf{I})/|\mathcal{N}(0,\mathbf{I})|$ and angle $\theta \sim \text{Uniform}(0, 2\pi)$, the rotation matrix is constructed as

$$\mathbf{R} = \mathbf{I} + \sin\theta[\mathbf{v}]_\times + (1-\cos\theta)[\mathbf{v}]_\times^2$$

where $[\mathbf{v}]_\times$ is the skew-symmetric matrix of $\mathbf{v}$. The same rotation is applied to both configurations: $\mathbf{x}'_{\text{low}} = \mathbf{R}\mathbf{x}_{\text{low}}$ and $\mathbf{x}'_{\text{high}} = \mathbf{R}\mathbf{x}_{\text{high}}$, preserving their geometric relationship while exposing the model to diverse orientations.

Variable molecular sizes are handled through strategic padding and masking within the data preprocessing pipeline. We extend smaller molecules to the maximum batch size $N_{\max}$ by inserting dummy atoms with negative type indices and zero coordinates. This creates a padding mask $M \in \{0,1\}^{B\times N_{\max}}$ where $M_{bi} = 1$ indicates a padded position. The transformer architecture leverages this mask for attention computation: masked positions receive attention logits of $-\infty$, yielding zero attention weights after softmax normalization. The graph architecture employs explicit exclusion by restricting neighborhood computations $\mathcal{N}_i$ to valid atoms only, while maintaining correct batch indexing through edge offset calculations.

3.6 Summary

We introduced neural-enhanced PT that addresses vanishing overlap by learning bijective transports T_θ between adjacent temperatures. The induced deterministic, involutive swap satisfies detailed balance and *can* potentially improve naive coordinate exchanges (measuring in terms of e.g. swap acceptance rates). We introduced three architectures spanning bias–capacity trade-offs, and clarifying their symmetry properties. Finally, we linked learning to practice by showing the bidirectional KL objective aligns with maximizing flow-enhanced acceptance, and we outlined a training pipeline tailored to molecular data.

Chapter 4

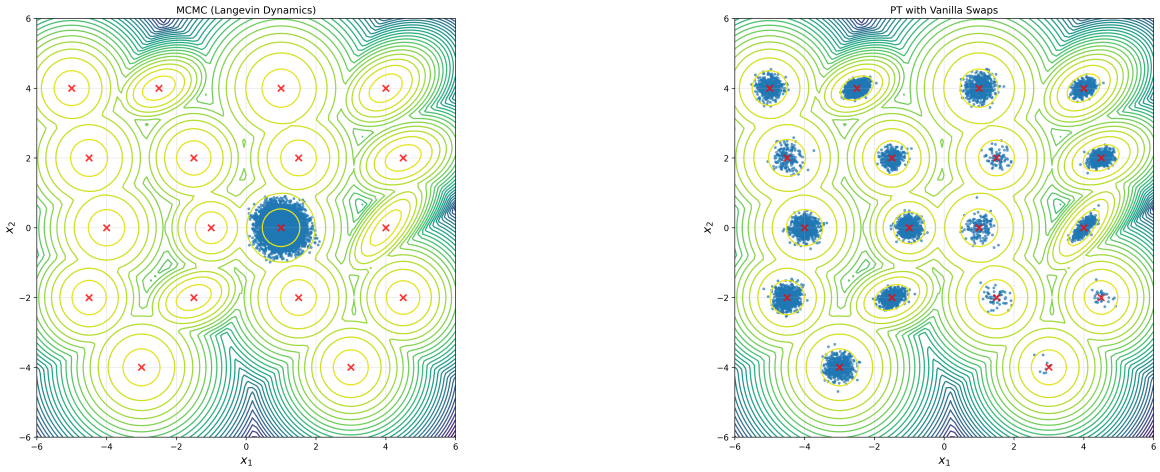
Experiments and Results

When you can measure and express in numbers, you know something;
when you cannot, your knowledge is meagre and unsatisfactory.
— William Thomson (1824–1907)

Having established the methodology, this chapter delivers an empirical test of whether learned transports can *construct* temperature overlap and translate it into faster molecular sampling. We benchmark our three architectures (introduced in 3.2) on dipeptides across a four-replica ladder. We open with an 18-mode GMM to visualise mode trapping and the dimensional decay of vanilla PT, then move to molecular datasets. Performance is assessed *mainly* via Swap Acceptance Rate (SAR), although we also provide further analysis on more topics.

4.1 Preliminaries

Before assessing flow-based acceleration on molecular systems, we first illustrate the limitations of MCMC samplers on a controlled benchmark: an 18-mode Gaussian mixture model (GMM) arranged on a 2D lattice with energy barriers between modes. We run overdamped Langevin dynamics with step size 0.01 for 10,000 iterations from a single mode, and compare against parallel tempering with two replicas at temperatures $T \in \{1.0, 20.0\}$, attempting swaps every 10 steps. Figure 4.1 summarizes the qualitative behavior: Langevin becomes trapped in a single mode, whereas vanilla PT can traverse between modes in this low-dimensional setting.



(a) Langevin sampling (mode trapping).

(b) PT with vanilla swaps.

Fig. 4.1 Sampling on an 18-mode GMM. **(a)** Langevin remains confined to one mode. **(b)** PT with coordinate swaps explores multiple modes in 2D.

However, the same vanilla PT mechanism deteriorates rapidly with dimension. In Figure 4.2 we report the mean swap-acceptance rate (accepted swaps divided by attempted swaps) as a function of the GMM dimension for several fixed temperature gaps. Acceptance remains reasonable in 2D but decays sharply as the dimension increases, dropping below 1% for moderate dimension. This mirrors the situation in molecular systems, where hundreds of degrees of freedom induce exceedingly sparse overlap between adjacent temperatures.

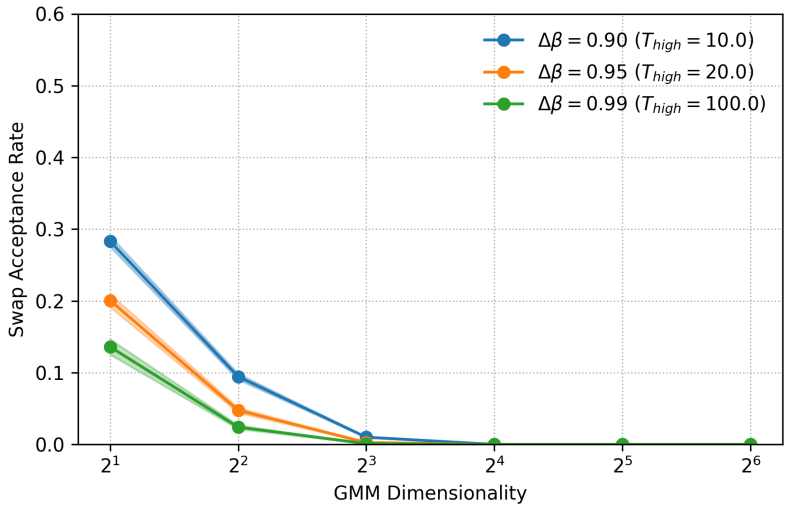


Fig. 4.2 SAR of vanilla PT versus GMM dimension for three temperature gaps.

These controlled experiments motivate our approach: while standard MCMC suffers mode trapping and vanilla PT alleviates this only in low dimensions, PT becomes inefficient as dimensionality grows because adjacent-temperature overlaps vanish. The goal of our learned transports is precisely to *construct* higher overlap, via energy so as to restore higher swap acceptance rates (SAR) in complex, high-dimensional molecular targets.

4.2 Dataset Generation

Our molecular datasets originate from PT simulations of nine dipeptide systems using the AMBER14 all-atom force field with implicit OBC2 solvent. The training set contains `[AA, SS, KK, AS, AK, SK]` peptides and test set `[SA, KA, KS]`. Each peptide begins from minimized PDB conformations obtained from the Timewarp 2AA-1-big repository (Klein, Foong, et al., 2023). After energy minimization to remove steric clashes, coordinates are perturbed by Gaussian noise ($\sigma = 0.01$ nm) to differentiate replicas. The geometric temperature ladder spans $T_1 = 300\,\mathrm{K}$ to $T_5 = 1000\,\mathrm{K}$ with scaling factor $r \approx 1.495$, yielding temperatures $[300.0, 450.0, 670.0, 1000.0]\,\mathrm{K}$ designed for $< 20\%$ naïve swap acceptance rate (SAR) (Table 4.1), leaving substantial room for improvement, yet avoiding the zero-SAR regime that would imply generating datasets via MALA.

The PT protocol employs Langevin dynamics with 10^{-4} nm step size, Metropolis-Hastings swap exchanges every 100 steps, and 5×10^6 total steps across 10 independent chains per temperature replica. From the resulting dataset, we later subsampled, retaining every *subsample_rate*–th sample, to reduce statistical correlation and improve computational efficiency. Energy evaluations are truncated at 10^8 kJ/mol with hard caps at 10^{20} kJ/mol to prevent numerical instabilities during training (as detailed in Section 3.5).

The preprocessing pipeline includes coordinate centering

$$\mathbf{r}'_i = \mathbf{r}_i - \frac{1}{N_a} \sum_{j=1}^{N_a} \mathbf{r}_j,$$

chirality filtering using ϕ–ψ dihedral angle criteria, and data augmentation via random rotations and translations. Trajectories are stored as tensors with shape $[N_T, N_c, N_s, 3N_a]$, where N_T is the number of temperature levels, N_c is the number of chains, N_s is the number of steps and $3N_a$ is the dimensionality of the molecule. We save PT trajectory alongside `atom_types.pt` and `adj_list.pt` (built via Equation (3.12)) files providing structural information for graph-aware architectures.

4.3 Swap Acceptance Rate (SAR)

We quantify swap efficiency for each adjacent temperature pair (β_k, β_{k+1}) by the *Swap Acceptance Rate* (SAR), defined as the expected Metropolis acceptance under the corresponding target(s). For the baseline (naive coordinate exchange), we use the acceptance $\alpha_{\text{naive}}(x,y)$ in (3.3). For the simple and graph flows (positions only), we use $\alpha_{\text{flow}}(x,y)$ in (3.6) with exponent $\Delta_{\text{flow}}(x,y)$ from (3.7). For the transformer (augmented states), we evaluate

$$\alpha_{\text{aug}}(x,y,v_x,v_y) \;=\; \min\{1, \exp(\Delta_{\text{aug}}(x,y,v_x,v_y))\},$$

with Δ_{aug} given in (3.24). In practice, given n pairs $\{(x_i,y_i)\}_{i=1}^n$ from $P_{\beta_k} \otimes P_{\beta_{k+1}}$, we estimate these with simple sample means. For positions-only methods (baseline, simple, graph),

$$\widehat{\text{SAR}}_{\text{baseline/flow}} = \frac{1}{n}\sum_{i=1}^{n} \alpha_{\text{naive/flow}}(x_i,y_i). \tag{4.1}$$

For the transformer,

$$\widehat{\text{SAR}}_{\text{transformer}} = \frac{1}{n}\sum_{i=1}^{n} \alpha_{\text{aug}}\big(x_i,y_i,v_{x,i},v_{y,i}\big), \qquad v_{x,i},v_{y,i} \sim \mathcal{N}(0,I),$$

For statistical rigor, we generally produce T independent repeats (distinct seeds/splits), then we report the mean $\bar{m} = \frac{1}{T}\sum_{t=1}^{T} m_t$ (m_t a per-repeat estimate) and a 95% Student t interval $\bar{m} \pm t_{0.975,T-1}\, s_m/\sqrt{T}$, where $s_m^2 = \frac{1}{T-1}\sum_{t=1}^{T}(m_t-\bar{m})^2$.

We first summarize baseline PT to establish difficulty across systems and temperature gaps. Table 4.1 shows uniformly modest acceptance, with a clear and expected size trend: larger peptides (e.g., 47-atom KK) accept less often than smaller ones (e.g., 23-atom AA/24-atom AS). We then turn to flow-enhanced PT on the adjacent pair $(k,k+1) = (0,1)$ (300–450 K), evaluating SAR with the acceptance functions above. Before discussing the numbers, we flag a key limitation: a late-stage HPC outage prevented the planned retraining from scratch. The results below were produced on very small datasets (via strongly subsampling) and short local CPU-only runs, so several models are undertrained. In this regime, simple (coordinate-only) flows appear particularly strong largely because they converge faster on single-peptide, vectorized domains; by contrast, multi-peptide settings enlarge the function class and slow optimization, and data-hungry transformers benefit less without scale. Graph flows outperform transformers here, which we attribute to their symmetry-aware inductive bias being more sample-efficient; transformers tend to realize their advantage only with larger datasets and longer training.

Table 4.1 Baseline Swap Acceptance Rates (%).

Peptide	Num Atoms	300–450K	450–670K	670–1000K
Training Systems				
AA	23	12.2 ± 0.2	19.4 ± 0.5	17.5 ± 0.4
AS	24	19.0 ± 0.3	15.7 ± 0.4	10.9 ± 0.2
AK	35	3.6 ± 0.2	9.2 ± 0.3	6.2 ± 0.2
SS	25	9.9 ± 0.2	9.9 ± 0.3	11.5 ± 0.2
KK	47	1.7 ± 0.2	3.9 ± 0.1	1.3 ± 0.1
SK	36	1.5 ± 0.1	8.9 ± 0.4	5.3 ± 0.2
Test Systems				
SA	24	13.3 ± 0.2	10.0 ± 0.4	10.4 ± 0.3
KA	35	6.0 ± 0.3	6.9 ± 0.2	8.6 ± 0.3
KS	36	3.3 ± 0.1	3.8 ± 0.2	6.4 ± 0.3

Table 4.2 SAR (%) by architecture for pair (300–450 K). Estimation follows (4.1)–(4.3).

System	Baseline PT	Simple Flow	Graph Flow	Transformer Flow
Training Systems				
AA	12.2 ± 0.2	82.2 ± 0.9	21.9 ± 0.4	16.3 ± 0.3
AS	19.0 ± 0.3	89.8 ± 1.2	26.0 ± 0.5	21.3 ± 0.6
AK	3.6 ± 0.2	89.3 ± 1.3	5.5 ± 0.3	4.7 ± 0.3
SS	9.9 ± 0.2	89.3 ± 0.5	10.7 ± 0.3	14.0 ± 0.3
KK	1.7 ± 0.2	89.2 ± 0.9	2.7 ± 0.2	3.1 ± 0.2
SK	1.5 ± 0.1	90.6 ± 1.0	2.9 ± 0.1	3.1 ± 0.1
Test Systems				
SA	13.3 ± 0.2	89.5 ± 0.7	16.3 ± 0.3	14.6 ± 0.3
KA	6.0 ± 0.3	89.2 ± 1.6	10.2 ± 0.3	7.4 ± 0.2
KS	3.3 ± 0.1	90.1 ± 1.2	7.3 ± 0.2	5.2 ± 0.1
Training Avg	**8.0**		**11.6**	**10.4**
Test Avg	**7.5**		**11.3**	**8.5**

Table 4.2 shows consistent improvements over baseline across learned transports. We highlight a promising indication of *transferability*. Recall that the Simple Flow model is trained separately on each peptide dataset, yielding nine distinct models (one per peptide). In contrast, the Graph Flow and Transformer models are trained once on the training systems and then evaluated directly on the test systems (peptides they have never encountered during training). The SAR results for these models therefore reflect genuine zero-shot transfer. Notably, this marks the first encouraging spark toward achieving our original goal: demonstrating transferability to unseen peptides. We further observe that the train–test performance gap is smaller for the graph-based architecture, which we attribute to its built-in symmetries that promote better generalization in low-data settings.

Across all methods, SAR declines with system size: even when transports improve overlap. Overall, these results should be read as preliminary and shaped by compute constraints; they nevertheless indicate that learned transports *can* elevate swap efficiency relative to baseline PT, while having exact sampling from our target.

4.4 Symmetry

Next, we examine whether the learned proposal $T_\theta : \mathbb{R}^{N\times 3} \to \mathbb{R}^{N\times 3}$ respects two basic symmetries on AA ($N = 23$) for the $(k,k+1) = (0,1)$ pair. As discussed in Section 4.3, the transformer was trained under tight budgets and without an explicit geometric bias, so its performance here reflects data- and scale-limited learning rather than a hard architectural limit. In contrast, the graph model encodes symmetry-aware operations by construction (see Section 3.2), and therefore provides a stronger reference on these metrics, as expected.

To probe $E(3)$ behavior, we sample i.i.d. rigid motions $(R,\mathbf{t})$ where $R \in \mathrm{SO}(3)$ (random axis–angle, uniform on the sphere) and $\mathbf{t} \in \mathbb{R}^3$ (uniform in a bounded cube), and for a batch $\{x_b\}_{b=1}^{B}$ measure the average per-atom discrepancy between "transform then map" and "map then transform":

$$\varepsilon_{\mathrm{E}(3)} \;=\; \frac{1}{BN}\sum_{b=1}^{B}\sum_{i=1}^{N}\big\| T_\theta(Rx_b+\mathbf{t})_i \;-\; (RT_\theta(x_b)+\mathbf{t})_i\big\|_2, \tag{4.2}$$

so a perfectly $E(3)$-equivariant T_θ yields $\varepsilon_{\mathrm{E}(3)} = 0$. We aggregate (4.2) over many random $(R,\mathbf{t})$ to obtain an empirical error distribution (Fig. 4.3). To assess permutation behavior, we construct random within-type atom permutations P (atoms permuted only inside their

element class) and compute

$$\varepsilon_{\text{perm}} = \frac{1}{BN} \sum_{b=1}^{B} \sum_{i=1}^{N} \left\| T_\theta(Px_b)_i - \left(PT_\theta(x_b)\right)_i \right\|_2, \tag{4.3}$$

again targeting zero in the ideal case; the resulting distribution appears in Fig. 4.4.

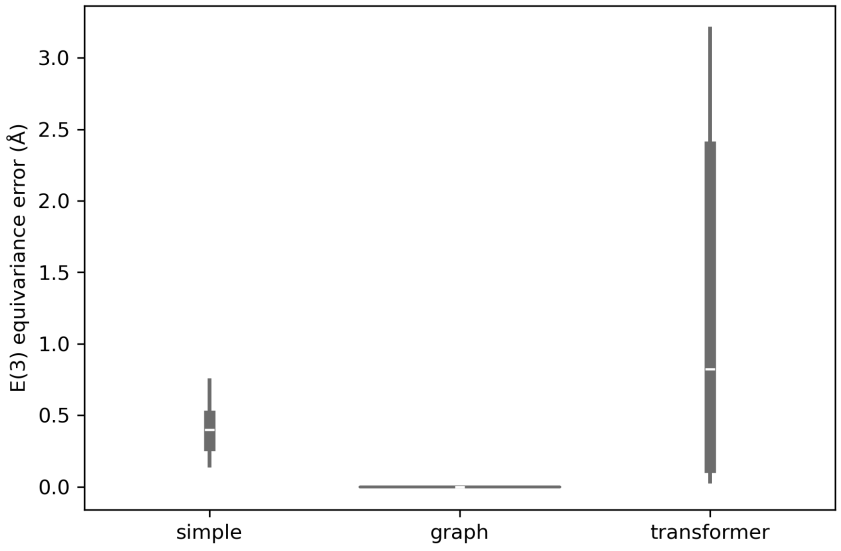


Fig. 4.3 Distribution of $\varepsilon_{\text{E(3)}}$ (Å) over random rigid motions.

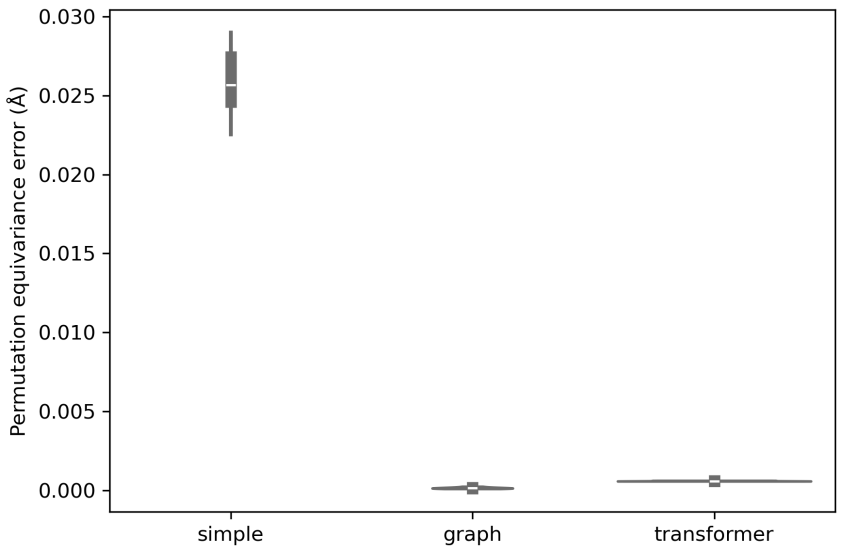


Fig. 4.4 Distribution of $\varepsilon_{\text{perm}}$ (Å) over random within-type permutations.

Taken together, the measurements align with the architectural biases. The graph flow attains near-zero medians on both metrics, consistent with its use of invariant scalars for scales and relative vectors for shifts, and with the centroid reparameterization that restores exact translation behavior discussed earlier. The simple coordinate-only flow, trained with rotation augmentation in low-data regime, partially preserves rigid motions but lacks a mechanism for atom relabeling, leading to noticeably larger $\varepsilon_{\text{perm}}$. The transformer shows near-permutation invariance (shared per-atom processing and masking help) yet a higher $\varepsilon_{\text{E(3)}}$ than the graph flow, which is expected in the absence of built-in $E(3)$ structure under limited training.

Because the transformer's symmetry arises purely from data augmentation, we anticipate substantial gains with longer training runs and larger datasets. Under stronger $E(3)$ augmentation applied consistently to (x, y) pairs, the transformer should close most of the $E(3)$ gap to the graph model while retaining its ability to capture long-range couplings at scale.

4.5 Attention and Adjacency

Next, we probe how `PTSwapTransformerFlow` routes information on AA (N=23) by extracting attention matrices across heads/layers and averaging over 320 configurations to obtain $A(i, j)$. Because raw attention is low-variance, we Z-score normalize to highlight relative structure. As "ground-truth" connectivity we use the covalent adjacency (with 22 bonds under the used neighborhood cut-off). We recall the graph model's neighborhoods are built as a radius graph with cutoff $\tau = 0.5\,\text{nm}$ (5 Å) when constructing edges (see 3.2).

The learned attention closely mirrors the ground–truth chemical bonds. As seen in Fig. 4.5, the transformer recovers the block–diagonal structure of bonded atom pairs with remarkable fidelity, with the strongest attention weights concentrated along existing bonds. Quantitatively, the Spearman correlation between attention scores and bond presence is high (ρ=0.812), and the Jensen–Shannon divergence between their distributions is low (0.215), indicating strong alignment. The *top-1* attended neighbor for each atom is always a bonded partner (100% recovery across all 23 atoms), and over 85% of total attention mass is assigned to bonded pairs. This shows the attention model organizes its attention to reflect the underlying molecular graph (with built-in topology in the graph-based model), providing intuition for how *inductive biases* toward graph structure can be advantageous.

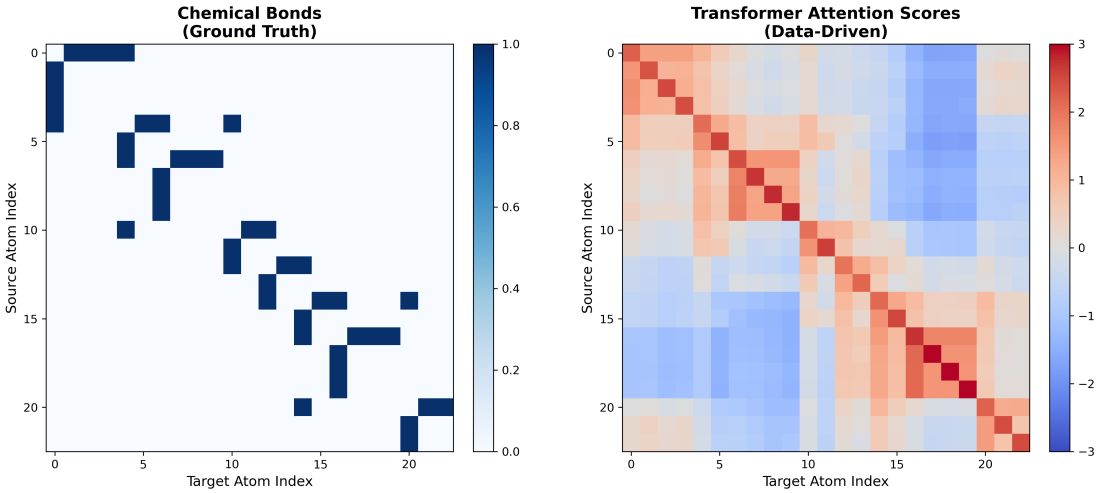


Fig. 4.5 AA bonds (left) vs. normalized transformer attention (right).

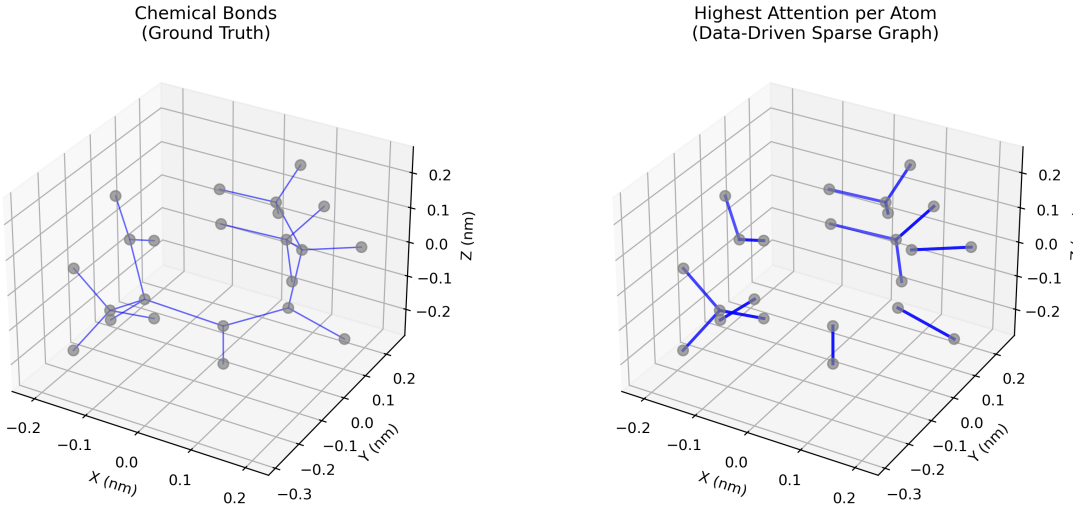


Fig. 4.6 3D bonds (left) vs. "attention skeleton" from top-1 neighbors (right).

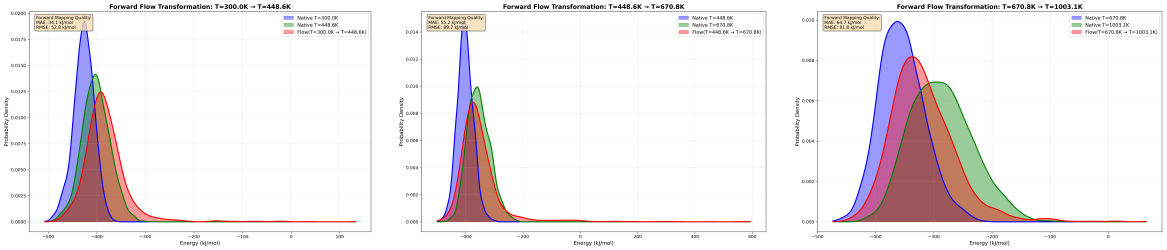
4.6 Energy Validation

We also validate *PTSwapFlow* (as it was the best-trained) T_θ on AA by checking whether it maps configurations from a lower to a higher temperature while preserving the *energy distribution*. For each adjacent pair in the ladder (300 K up to 1000 K), we draw M=2000 configurations from the lower replica $x^{(i)}_{\text{low}}$ and from the higher replica $y^{(i)}_{\text{high}}$, compute mapped states $x^{(i)}_{\text{mapped}} = T_\theta\left(x^{(i)}_{\text{low}}\right)$, and compare potential energies under the common $U(\cdot)$. As a

paired summary error we report the mean absolute error

$$\mathrm{MAE} = \frac{1}{M}\sum_{i=1}^{M}\left|U\left(x_{\mathrm{mapped}}^{(i)}\right) - U\left(y_{\mathrm{high}}^{(i)}\right)\right|, \tag{4.4}$$

and visualize overlap using Gaussian–KDE curves for native low-temperature (blue), low-to-high mapped (red) and native high-temperature (green) energies.



(a) 300 K → 450 K, MAE = 34.1 kJ/mol (b) 450 K → 670 K, MAE = 55.2 kJ/mol (c) 670 K → 1000 K, MAE = 64.7 kJ/mol

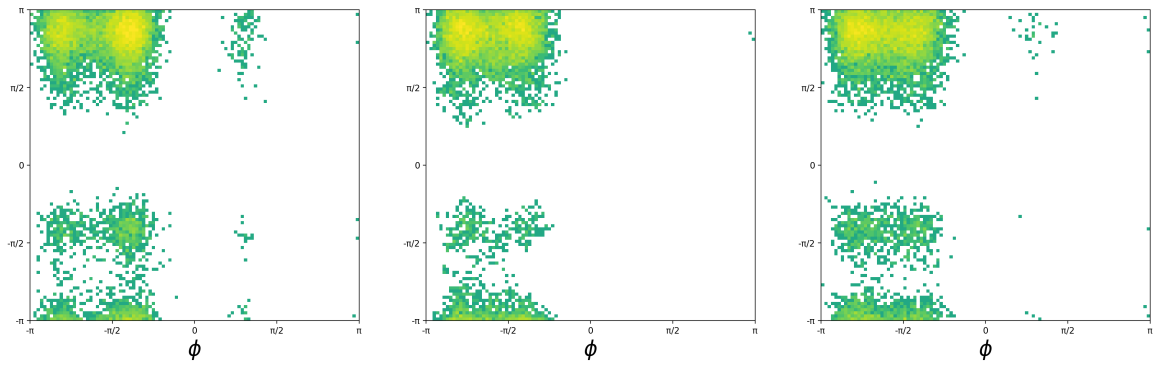
Fig. 4.7 Energy distributions for AA under the simple-flow model across the three adjacent temperature pairs (300 → 450 K, 450 → 670 K, and 670 → 1000 K). Densities are estimated via Gaussian KDE. The x-axis denotes energy (kJ mol^{-1}) and the y-axis denotes density.

Across these pairs, conservation quality degrades as the temperature gap widens. The 300→450 K transition shows better overlap (MAE ≈ 34 kJ/mol); the next two remain in a good–moderate regime (MAE ≈ 55–65 kJ/mol) with small but noticeable shifts. Thus, for AA, the simple flow preserves energy distributions well for small to moderate gaps, supporting its use for accelerating swaps on the cooler end of the ladder. Figure 4.7 illustrates the idea of *constructing overlap*, leveraging the fact that the probability distribution is directly tied to the energy function.

4.7 Ramachandran Coverage

Finally, we conclude with an end–to–end sampling test on AA using Ramachandran (ϕ, ψ) statistics (Ramachandran et al., 1963). As a *ground truth* reference we use the Timewarp `2AA-1-complete` MD dataset, which provides all-atom trajectories for 400 dipeptides (classical MD with an implicit solvent; frames saved every 10,000 steps). We compare (i) *vanilla* PT on a four-replica ladder $T \in \{300, 450, 670, 1000\}$ K with naïve coordinate swaps (Section 4.3), and (ii) *flow–PT* using the simple architecture. For flow–PT we train one map per adjacent pair, $\{T_{0\to1}, T_{1\to2}, T_{2\to3}\}$, and at each exchange attempt between replicas k and $k+1$ we apply the corresponding map and test with the Metropolis rule in (3.6). Since HPC

downtime forced CPU-only training; compute-heavy graph/transformer transports underfit, yielding only 2–10% SAR gains as shown in (Sec. 4.3) and Ramachandran coverage is comparable to vanilla PT. We therefore omit their plots and focus on the simple flow, which trained more reliably under the same constraints, with 70–90% SAR gains.



(a) MD ground truth. (b) Vanilla PT. (c) Flow–PT.

Fig. 4.8 Ramachandran plots for AA at the 300 K replica after running PT in a four–temperature (geom schedule) ladder (300–1000 K).

Figure 4.8 summarizes the qualitative picture. Vanilla PT remains confined to the high-density basins and struggles to reach smaller, low-density modes, suggesting limited guidance from higher-temperature replicas to explore those regions. Flow-based PT remedies this limitation, broadening coverage and placing samples in modes that vanilla PT does not visit. This indicates that the learned transports are effectively *constructing* temperature overlap and converting it into successful exchanges, thereby enabling more thorough exploration of the target distribution.

4.8 Summary

We show empirically that learned transports can *construct* temperature overlap and lift swap acceptance beyond baseline PT, which shows the expected size dependence (larger systems $\Rightarrow$ lower SAR) formulated in the *temperature overlap problem* formulated in previous chapter. Under tight budgets, simple flows look strongest (fast convergence on single-peptide domains), while graph flows provide a robust compromise; solid SAR with near-ideal symmetry. The transformer already learns useful long-range couplings (attention–adjacency) but needs more training to reduce $E(3)$ error and close the gap. Energy checks on AA show the simple flow preserves high-temperature energy distributions for moderate gaps. These results are preliminary; the path forward is to scale training (GPUs, longer runs).

Chapter 5

Conclusion

The purpose of computing is insight, not numbers.
— Richard Hamming (1915–1998)

This thesis advances a simple but powerful idea: instead of hoping adjacent temperatures accidentally overlap, we can engineer that overlap with exact, learned transports, and we can do so in a way that *transfers across molecules and temperature ladders*. Starting from statistical–mechanical first principles (Boltzmann equilibrium) through the limits of local MCMC and the ensemble logic of PT, we identified a true bottleneck: in high dimensions, neighboring tempered distributions rarely meet. Our solution was to learn diffeomorphic, symmetry-aware maps that actively warp probability geometry so that replica exchanges become more common while detailed balance is preserved via an involutive swap. Crucially, we framed this as more than acceleration: physically similar systems share structure, and samplers should exploit that to propose in a principled way wherever they are placed on a ladder. Normalizing flows give the reversible mechanism; graph and transformer variants supply inductive bias and global coupling; and the bidirectional NLL directly aligns learning with the Metropolis exponent. On dipeptides, this translated into higher swap acceptance and broader configurational coverage at the physical replica.

5.1 Limitations

A central architectural limitation is that we currently train one model per adjacent pair: an N-replica ladder requires $N-1$ distinct flows. Ideally, temperature would be part of the input so a single conditional transport $T_\theta(x \mid \beta_{\text{in}}, \beta_{\text{out}})$ serves the entire ladder; our present methods do not yet support this.

Empirically, our strongest results focus on swap acceptance rate (SAR) under constrained training. We did not conduct extensive ablations, nor did we pursue joint ladder co-optimization once SAR gains failed to materialize robustly in the potentially transferable models (graph and transformers). Evaluation breadth is therefore limited: beyond SAR we did not focus on other end-to-end measures (e.g., ESS per wall-clock, round-trip times), and we should do so in future work.

Although compute constraints did play a role in (harming) the outcome of this work, transferable models were trained on local CPUs for $> 24\,\mathrm{h}$; and so optimization bottlenecks might also have to do coupling-based flows, and not only with scale (e.g. training with more data and compute). We have not yet tested higher-capacity alternatives such as autoregressive flows, and residual/continuous-time layers variants outlined in §2.3.2, and we should do so in future work.

5.2 Future Work

Scale is the most direct next step: retrain with GPUs, longer schedules, and larger datasets to realize transformer capacity and stress-test transferability (which we expect to improve to some degree). In parallel, embed temperature directly into the transport to collapse $N-1$ models into a single conditional map, and consider co-optimizing the ladder jointly with the learned proposal.

Architecturally, explore $E(3)$/SE(3)-equivariant variants, autoregressive normalizing flows, and residual/continuous-time layers, as well as diffusion or flow-matching objectives as alternatives to pure likelihood training. Light acceptance-aware or equivariance regularizers that target inference-time quantities are natural complements.

Evaluation should broaden beyond SAR: report ESS per unit time/energy, replica round-trip rates, backbone and side-chain statistics, free-energy landscapes, and compare against stronger PT baselines so improvements reflect true end-to-end efficiency. In this work, we did not emphasize stronger benchmarking here because only the non-transferable "Simple Flow" reached considerable outperformance and so we spent more time on improving the models under the compute constrains. This work then effectively replicates (Zhang et al., 2025b) work, which is still valuable to the field, but the next iteration should prioritize ladder-agnostic and peptide transfer with rigorous efficiency accounting.

References

Akhound-Sadegh, Tara et al. (2025). *Progressive Inference-Time Annealing of Diffusion Models for Sampling from Boltzmann Densities*. arXiv: 2506.16471 [cs.LG]. URL: https://arxiv.org/abs/2506.16471.

Aldous, David and James Allen Fill (2002). *Reversible Markov Chains and Random Walks on Graphs*. Unfinished monograph; recompiled PDF (2014). URL: https://www.stat.berkeley.edu/~aldous/RWG/book.html.

Banach, Stefan (1922). "Sur les opérations dans les ensembles abstraits et leur application aux équations intégrales". In: *Fundamenta Mathematicae* 3, pp. 133–181. DOI: 10.4064/fm-3-1-133-181.

Behrmann, Jens et al. (2019). *Invertible Residual Networks*. arXiv: 1811.00995 [cs.LG]. URL: https://arxiv.org/abs/1811.00995.

Bellman, Richard E. (1957). *Dynamic Programming*. Princeton, NJ: Princeton University Press.

Boltzmann, Ludwig (1868). "Studien über das Gleichgewicht der lebendigen Kraft zwischen bewegten materiellen Punkten". German. In: *Sitzungsberichte der k. k. Akademie der Wissenschaften (Wien), II. Klasse* 58, pp. 517–560.

Chen, Ricky T. Q. et al. (2019). *Neural Ordinary Differential Equations*. arXiv: 1806.07366 [cs.LG]. URL: https://arxiv.org/abs/1806.07366.

Dinh, Laurent, Jascha Sohl-Dickstein, and Samy Bengio (2017). "Density Estimation using Real NVP". In: *International Conference on Learning Representations*. arXiv: 1605.08803.

Douc, Randal and Sylvain Le Cor (2015). *Markov Chain Monte Carlo – Theory and Practical Applications: A Crash Course in Monte-Carlo Methods by Markov Chains*. https://wiki.randaldouc.xyz/lib/exe/fetch.php?media=world:poly.pdf. Graduate lecture notes, Télécom SudParis.

Draxler, Felix, Peter Sorrenson, et al. (2023). "Free-Form Flows: Make Any Architecture a Normalizing Flow". In: DOI: 10.48550/arXiv.2310.16624. arXiv: 2310.16624 [cs.LG].

Draxler, Felix, Stefan Wahl, et al. (2024). "On the Universality of Volume-Preserving and Coupling-Based Normalizing Flows". In: DOI: 10.48550/arXiv.2402.06578. arXiv: 2402.06578 [cs.LG].

Flouris, Kyriakos and Ender Konukoglu (2023). "Canonical Normalizing Flows for Manifold Learning". In: DOI: 10.48550/arXiv.2310.12743. arXiv: 2310.12743 [stat.ML].

Grathwohl, Will et al. (2018). *FFJORD: Free-form Continuous Dynamics for Scalable Reversible Generative Models*. arXiv: 1810.01367 [cs.LG]. URL: https://arxiv.org/abs/1810.01367.

Hamilton, William Rowan (1834). "On a General Method in Dynamics". In: *Philosophical Transactions of the Royal Society of London* 124, pp. 247–308. DOI: 10.1098/rstl.1834.0017.

He, Jiajun et al. (2025). "No Trick, No Treat: Pursuits and Challenges Towards Simulation-Free Training of Neural Samplers". In: DOI: 10.48550/arXiv.2502.06685. arXiv: 2502.06685 [cs.LG].

Ho, Jonathan, Ajay Jain, and Pieter Abbeel (2020). *Denoising Diffusion Probabilistic Models*. arXiv: 2006.11239 [cs.LG]. URL: https://arxiv.org/abs/2006.11239.

Kingma, Diederik P. and Jimmy Ba (2017). *Adam: A Method for Stochastic Optimization*. arXiv: 1412.6980 [cs.LG]. URL: https://arxiv.org/abs/1412.6980.

Kingma, Diederik P. and Prafulla Dhariwal (2018). "Glow: Generative Flow with Invertible 1×1 Convolutions". In: *Advances in Neural Information Processing Systems*. arXiv: 1807.03039.

Kingma, Diederik P., Tim Salimans, et al. (2017). *Improving Variational Inference with Inverse Autoregressive Flow*. arXiv: 1606.04934 [cs.LG]. URL: https://arxiv.org/abs/1606.04934.

Klein, Leon, Andrew Y. K. Foong, et al. (2023). "Timewarp: Transferable Acceleration of Molecular Dynamics by Learning Time-Coarsened Dynamics". In: DOI: 10.48550/arXiv.2302.01170. arXiv: 2302.01170 [stat.ML].

Klein, Leon, Andreas Krämer, and Frank Noé (2023). "Equivariant Flow Matching". In: DOI: 10.48550/arXiv.2306.15030. arXiv: 2306.15030 [stat.ML].

Klein, Leon and Frank Noé (2024). "Transferable Boltzmann Generators". In: DOI: 10.48550/arXiv.2406.14426. arXiv: 2406.14426 [stat.ML].

Kramers, H. A. (1940). "Brownian Motion in a Field of Force and the Diffusion Model of Chemical Reactions". In: *Physica* 7.4, pp. 284–304. DOI: 10.1016/S0031-8914(40)90098-2.

Kullback, S. and R. A. Leibler (1951). "On Information and Sufficiency". In: *The Annals of Mathematical Statistics* 22.1, pp. 79–86. DOI: 10.1214/aoms/1177729694. URL: https://doi.org/10.1214/aoms/1177729694.

Langevin, Paul (1908). "On the Theory of Brownian Motion". In: *Comptes Rendus* 146, pp. 530–533.

Lee, Sangyun, Zinan Lin, and Giulia Fanti (2024). "Improving the Training of Rectified Flows". In: DOI: 10.48550/arXiv.2405.20320. arXiv: 2405.20320 [cs.LG].

Lipman, Yaron et al. (2023). *Flow Matching for Generative Modeling*. arXiv: 2210.02747 [cs.LG]. URL: https://arxiv.org/abs/2210.02747.

Papamakarios, George, Theo Pavlakou, and Iain Murray (2018). *Masked Autoregressive Flow for Density Estimation*. arXiv: 1705.07057 [stat.ML]. URL: https://arxiv.org/abs/1705.07057.

Predescu, Cristian, Mihaela Predescu, and Cristian V. Ciobanu (2004). "The incomplete beta function law for parallel tempering sampling of classical canonical systems". In: *The Journal of Chemical Physics* 120.9, pp. 4119–4128. DOI: 10.1063/1.1644093.

Rahimi, Ali and Benjamin Recht (2007). "Random Features for Large-Scale Kernel Machines". In: *Advances in Neural Information Processing Systems*. Ed. by J. Platt et al. Vol. 20. Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper_files/paper/2007/file/013a006f03dbc5392effeb8f18fda755-Paper.pdf.

Ramachandran, G. N., C. Ramakrishnan, and V. Sasisekharan (1963). "Stereochemistry of Polypeptide Chain Configurations". In: *Journal of Molecular Biology* 7, pp. 95–99. DOI: 10.1016/S0022-2836(63)80023-6.

Risken, Hannes (1996). *The Fokker–Planck Equation: Methods of Solution and Applications*. Springer.

Rissanen, Severi et al. (2025). *Progressive Tempering Sampler with Diffusion*. arXiv: 2506.05231 [cs.LG]. URL: https://arxiv.org/abs/2506.05231.

Schreiner, Mathias, Ole Winther, and Simon Olsson (2023). "Implicit Transfer Operator Learning: Multiple Time-Resolution Models for Molecular Dynamics". In: *Advances in Neural Information Processing Systems 37*. OpenReview ID: 1kZx7JiuA2. URL: https://openreview.net/forum?id=1kZx7JiuA2.

Tan, Charlie B., Avishek Joey Bose, et al. (2025). "Scalable Equilibrium Sampling with Sequential Boltzmann Generators". In: DOI: 10.48550/arXiv.2502.18462. arXiv: 2502.18462 [stat.ML].

Tan, Charlie B., Majdi Hassan, et al. (2025). "Amortized Sampling with Transferable Normalizing Flows". In: *ICML 2025 Workshop on Generative Models for Biology*. OpenReview ID: 1nNvu3hJqP. URL: https://openreview.net/forum?id=1nNvu3hJqP.

Vaswani, Ashish et al. (2023). *Attention Is All You Need*. arXiv: 1706.03762 [cs.CL]. URL: https://arxiv.org/abs/1706.03762.

Yu, Ziyang, Wenbing Huang, and Yang Liu (2024). "Force-Guided Bridge Matching for Full-Atom Time-Coarsened Dynamics of Peptides". In: DOI: 10.48550/arXiv.2408.15126. arXiv: 2408.15126 [physics.chem-ph].

Zhai, Shuangfei et al. (2024). "Normalizing Flows are Capable Generative Models". In: DOI: 10.48550/arXiv.2412.06329. arXiv: 2412.06329 [cs.LG].

Zhang, Leo et al. (2025a). "Accelerated Parallel Tempering via Neural Transports". In: DOI: 10.48550/arXiv.2502.10328. arXiv: 2502.10328 [stat.ML].

— (May 2025b). "Accelerated Parallel Tempering via Neural Transports". In: *arXiv preprint arXiv:2502.10328v2*. Under review.

Appendix A

Proofs & Supplementary Mathematics

A.1 MH kernel satisfies detailed balance

This appendix verifies that the Metropolis–Hastings (MH) transition kernel from Equation (2.9) satisfies detailed balance with respect to any target density π.

Proof. Consider off-diagonal pairs $\boldsymbol{x \neq y}$, since the argument for diagonal pairs $\boldsymbol{y = x}$ is trivial. We prove the pairwise flux symmetry

$$\pi(x)\, q(y \mid x)\, \alpha(x,y) = \pi(y)\, q(x \mid y)\, \alpha(y,x).$$

Consider two cases for $r(x,y)$.

Case 1: $r(x,y) \geq 1$. Then $\alpha(x,y) = 1$ and $\alpha(y,x) = \min\{1, 1/r(x,y)\} = 1/r(x,y)$. Hence

$$\begin{aligned} \pi(x)\, q(y \mid x)\, \alpha(x,y) &= \pi(x)\, q(y \mid x) \\ &= \frac{\pi(y)\, q(x \mid y)}{r(x,y)} = \pi(y)\, q(x \mid y)\, \alpha(y,x). \end{aligned}$$

Case 2: $r(x,y) \leq 1$. Then $\alpha(x,y) = r(x,y)$ and $\alpha(y,x) = 1$. Hence

$$\begin{aligned} \pi(x)\, q(y \mid x)\, \alpha(x,y) &= \pi(x)\, q(y \mid x)\, r(x,y) \\ &= \pi(y)\, q(x \mid y) = \pi(y)\, q(x \mid y)\, \alpha(y,x). \end{aligned}$$

Thus $\pi(x)K(x,y) = \pi(y)K(y,x)$ for all $x \neq y$, i.e., detailed balance holds. □

A.2 2D Affine Coupling Layer Example

This appendix provides a concrete 2D example demonstrating the mechanics of affine coupling layers, including forward transformations, inverse computations, and Jacobian determinant calculations.

Consider a 2D input vector $\mathbf{x}^{(0)} = (x_1, x_2)$ processed through two sequential affine coupling layers with alternating masks:

$$\left.\begin{array}{l} \mathbf{m}^{(1)} = (1,0) \\ \mathbf{m}^{(2)} = (0,1) \end{array}\right\} \quad \begin{array}{l} \mathbf{x}^{(0)} = (x_1, x_2) \xrightarrow{\mathbf{x}^{(0)} \odot \mathbf{m}^{(1)}} (x_1, 0) \\ \mathbf{x}^{(1)} = (y_1, y_2) \xrightarrow{\mathbf{x}^{(1)} \odot \mathbf{m}^{(2)}} (0, y_2) \end{array}$$

The transformation proceeds through two coupling layers as detailed in Table A.1.

Table A.1 Forward coupling layer transformations for 2D example.

Layer	**Mask m**	**Forward transformation**	$\log \lvert \det J \rvert$
1	$(1,0)$	$y_1 = x_1,\ y_2 = x_2 e^{s^{(1)}(x_1)} + t^{(1)}(x_1)$	$s^{(1)}(x_1)$
2	$(0,1)$	$z_2 = y_2,\ z_1 = y_1 e^{s^{(2)}(y_2)} + t^{(2)}(y_2)$	$s^{(2)}(y_2)$

The full Jacobian matrices for each layer are given by

$$J^{(1)} = \frac{\partial (y_1, y_2)}{\partial (x_1, x_2)} = \begin{pmatrix} 1 & 0 \\ \frac{\partial}{\partial x_1}\left[x_2 e^{s^{(1)}(x_1)} + t^{(1)}(x_1)\right] & e^{s^{(1)}(x_1)} \end{pmatrix}$$

$$J^{(2)} = \frac{\partial (z_1, z_2)}{\partial (y_1, y_2)} = \begin{pmatrix} e^{s^{(2)}(y_2)} & y_1 e^{s^{(2)}(y_2)} s^{(2)\prime}(y_2) + t^{(2)\prime}(y_2) \\ 0 & 1 \end{pmatrix}$$

The composite forward map $f(\mathbf{x}) = \mathbf{z}$ has log-determinant

$$\log \lvert \det J_f(\mathbf{x}) \rvert = s^{(1)}(x_1) + s^{(2)}(y_2)$$

where $y_2 = x_2 e^{s^{(1)}(x_1)} + t^{(1)}(x_1)$. The inverse transformation proceeds by reversing the layer order and applying inverse operations, as shown in Table A.2.

Table A.2 Inverse coupling layer transformations for 2D example.

Layer	**Mask m**	**Inverse transformation**	$\log\lvert\det J\rvert$
2^{-1}	$(0,1)$	$y_2 = z_2,\ y_1 = \left(z_1 - t^{(2)}(z_2)\right) e^{-s^{(2)}(z_2)}$	$-s^{(2)}(z_2)$
1^{-1}	$(1,0)$	$x_1 = y_1,\ x_2 = \left(y_2 - t^{(1)}(y_1)\right) e^{-s^{(1)}(y_1)}$	$-s^{(1)}(y_1)$

The composite inverse map $f^{-1}(\mathbf{z}) = \mathbf{x}$ has log-determinant:

$$\log\lvert\det J_{f^{-1}}(\mathbf{z})\rvert = -s^{(2)}(z_2) - s^{(1)}(y_1)$$

where $y_1 = \left(z_1 - t^{(2)}(z_2)\right) e^{-s^{(2)}(z_2)}$. This example demonstrates that coupling layers achieve exact invertibility and tractable Jacobians through alternating masks. The log-determinant sums across layers: $\log\lvert\det J_f\rvert = \sum_i \log\lvert\det J^{(i)}\rvert$, enabling computationally efficient normalizing flows.

A.3 MH acceptance rate with deterministic proposal

This appendix derives the MH acceptance criterion for deterministic proposals (3.5).

Proof. Let $\pi(x,y)$ be the target density on $\Omega\times\Omega$ with respect to Lebesgue measure $\lambda\otimes\lambda$. Consider a deterministic proposal induced by a measurable map $g:\Omega\times\Omega\to\Omega\times\Omega$,

$$q\big((x,y),\mathrm{d}(x',y')\big) \;=\; \delta_{g(x,y)}\big(\mathrm{d}(x',y')\big),$$

which is singular w.r.t. $\lambda\otimes\lambda$. The Metropolis–Hastings transition kernel is

$$K\big((x,y),\mathrm{d}(x',y')\big) \;=\; \alpha(x,y)\,\delta_{g(x,y)}\big(\mathrm{d}(x',y')\big) \;+\; \big(1-\alpha(x,y)\big)\,\delta_{(x,y)}\big(\mathrm{d}(x',y')\big).$$

Detailed balance requires, as an identity of measures,

$$\pi(x,y)\,K\big((x,y),\mathrm{d}(x',y')\big) \;=\; \pi(x',y')\,K\big((x',y'),\mathrm{d}(x,y)\big).$$

Equivalently, for any bounded test function φ,

$$\begin{aligned}\int \pi(x,y)\big[\alpha(x,y)\,\varphi\big(g(x,y)\big)+\big(1-\alpha(x,y)\big)\,\varphi(x,y)\big]\,\mathrm{d}x\,\mathrm{d}y &= \int \pi(x,y)\,\varphi(x,y)\,\mathrm{d}x\,\mathrm{d}y\\ \Longrightarrow\quad \int \pi(x,y)\,\alpha(x,y)\,\varphi\big(g(x,y)\big)\,\mathrm{d}x\,\mathrm{d}y &= \int \pi(x,y)\,\alpha(x,y)\,\varphi(x,y)\,\mathrm{d}x\,\mathrm{d}y.\end{aligned}$$

Apply the change of variables $(u,v)=g(x,y)$ (so $(x,y)=g(u,v)$). If g is a C^1 diffeomorphism (in particular, an involution), then $\mathrm{d}x\,\mathrm{d}y \;=\; \big|\det J_g(u,v)\big|\,\mathrm{d}u\,\mathrm{d}v$, and the left-hand side becomes

$$\int \pi\big(g(u,v)\big)\,\alpha\big(g(u,v)\big)\,\varphi(u,v)\,\big|\det J_g(u,v)\big|\,\mathrm{d}u\,\mathrm{d}v.$$

Since the equality holds for all φ, we must have for a.e. (u,v), $\pi\big(g(u,v)\big)\,\alpha\big(g(u,v)\big)\,\big|\det J_g(u,v)\big| = \pi(u,v)\,\alpha(u,v)$. Renaming $(u,v)\mapsto(x,y)$ gives the symmetry condition

$$\pi(x,y)\,\alpha(x,y) \;=\; \pi\big(g(x,y)\big)\,\alpha\big(g(x,y)\big)\,\big|\det J_g(x,y)\big|. \tag{A.1}$$

A standard solution of (A.1) is the Metropolis rule

$$\alpha(x,y) \;=\; \min\left\{1,\; \frac{\pi\big(g(x,y)\big)\,\big|\det J_g(x,y)\big|}{\pi(x,y)}\right\}, \tag{A.2}$$

which is exactly (3.5). □

A.4 Flow-Enhanced Acceptance Criterion

This appendix derives the flow-enhanced acceptance criterion Δ_{flow} used in our flow-based PT swap proposals (3.7).

Proof. For a dimension-preserving, differentiable, *involutive* deterministic proposal g (i.e. $g \circ g = \mathrm{id}$), the MH ratio is

$$R(x,y) \;=\; \frac{\pi(g(x,y))\,\big|\det J_g(x,y)\big|}{\pi(x,y)}.$$

as shown in A.3. The Jacobian of g has block form

$$J_g(x,y) = \begin{bmatrix} 0 & J_{T_\theta^{-1}}(y) \\ J_{T_\theta}(x) & 0 \end{bmatrix}, \qquad \Rightarrow \qquad \big|\det J_g(x,y)\big| = \big|\det J_{T_\theta}(x)\big|\,\big|\det J_{T_\theta^{-1}}(y)\big|,$$

(the swap contributes only a sign, removed by the absolute value). Therefore

$$R(x,y) = \frac{p_{\text{low}}(T_\theta^{-1}(y))\,p_{\text{high}}(T_\theta(x))}{p_{\text{low}}(x)\,p_{\text{high}}(y)} \cdot \big|\det J_{T_\theta}(x)\big|\,\big|\det J_{T_\theta^{-1}}(y)\big|.$$

Taking logs yields Δ_{flow} from Equation (**??**). For Boltzmann densities, the normalisers cancel and

$$\log \frac{p_{\text{low}}(T_\theta^{-1}(y))\,p_{\text{high}}(T_\theta(x))}{p_{\text{low}}(x)\,p_{\text{high}}(y)} = -\beta_{\text{low}}U(T_\theta^{-1}(y)) - \beta_{\text{high}}U(T_\theta(x)) + \beta_{\text{low}}U(x) + \beta_{\text{high}}U(y),$$

which gives Equation (3.7). □

A.5 Proof of Lemma 3.1.1

This appendix proves Lemma 3.1.1, i.e., that the Metropolis–Hastings kernel with the deterministic involutive proposal $(x,y) \mapsto g(x,y)$ satisfies detailed balance with respect to the product target density $\pi(x,y) = p_{\beta_k}(x)\, p_{\beta_{k+1}}(y)$.

Proof. Fix $(x,y) \in \Omega \times \Omega$ and write $(x',y') = g(x,y)$. Off the diagonal $\big((x',y') \neq (x,y)\big)$, the deterministic proposal moves only along the pair $(x,y) \leftrightarrow (x',y')$, so it suffices to verify the pairwise flux symmetry

$$\pi(x,y)\,\alpha(x,y) \;=\; \pi(x',y')\,\big|\det J_g(x,y)\big|\,\alpha(x',y').$$

Define the local ratio

$$r(x,y) \;:=\; \frac{\pi(x',y')\,\big|\det J_g(x,y)\big|}{\pi(x,y)}.$$

Because g is an involution and $J_g(x',y')$ is the inverse of $J_g(x,y)$, we have $r(x',y') = 1/r(x,y)$. The Metropolis rule is

$$\alpha(x,y) = \min\{1,\, r(x,y)\}, \qquad \alpha(x',y') = \min\{1,\, r(x',y')\} = \min\{1,\, 1/r(x,y)\}.$$

Case 1: $r(x,y) \geq 1$. Then $\alpha(x,y) = 1$ and $\alpha(x',y') = 1/r(x,y)$, so

$$\pi(x,y)\,\alpha(x,y) = \pi(x,y) = \pi(x',y')\,\big|\det J_g(x,y)\big|\,\frac{1}{r(x,y)} = \pi(x',y')\,\big|\det J_g(x,y)\big|\,\alpha(x',y').$$

Case 2: $r(x,y) \leq 1$. Then $\alpha(x,y) = r(x,y)$ and $\alpha(x',y') = 1$, so

$$\pi(x,y)\,\alpha(x,y) = \pi(x,y)\,r(x,y) = \pi(x',y')\,\big|\det J_g(x,y)\big| = \pi(x',y')\,\big|\det J_g(x,y)\big|\,\alpha(x',y').$$

Thus the off-diagonal fluxes match. The diagonal part $\big((x',y') = (x,y)\big)$ is trivially symmetric, hence detailed balance holds with respect to π. □

A.6 KL ↔ NLL (positions only).

Write the Boltzmann densities as $p_\beta(x) = Z_\beta^{-1} e^{-\beta U(x)}$. For an invertible $T_\theta : \Omega \to \Omega$, the pushforwards have densities

$$
\begin{aligned}
q_\theta(y) &\equiv (T_\theta \sharp P_{\beta_k})(\mathrm{d}y)/\mathrm{d}y = p_{\beta_k}\big(T_\theta^{-1}(y)\big)\,\big|\det J_{T_\theta^{-1}}(y)\big|, \\
r_\theta(x) &\equiv (T_\theta^{-1} \sharp P_{\beta_{k+1}})(\mathrm{d}x)/\mathrm{d}x = p_{\beta_{k+1}}\big(T_\theta(x)\big)\,\big|\det J_{T_\theta}(x)\big|.
\end{aligned}
$$

By definition and separating out θ–independent parts, we derive the rirst KL term as follows

$$
\begin{aligned}
D_{\mathrm{KL}}\big(P_{\beta_{k+1}} \,\|\, T_\theta \sharp P_{\beta_k}\big) &= \int p_{\beta_{k+1}}(y)\,\log \frac{p_{\beta_{k+1}}(y)}{q_\theta(y)}\,\mathrm{d}y \\
&= \underbrace{\int p_{\beta_{k+1}}(y)\,\log p_{\beta_{k+1}}(y)\,\mathrm{d}y}_{\text{const. w.r.t. } \theta} - \int p_{\beta_{k+1}}(y)\,\log q_\theta(y)\,\mathrm{d}y \\
&= \text{const.} - \mathbb{E}_{y\sim P_{\beta_{k+1}}}\Big[\log p_{\beta_k}\big(T_\theta^{-1}(y)\big) + \log\big|\det J_{T_\theta^{-1}}(y)\big|\Big] \\
&= \text{const.} - \mathbb{E}_{y\sim P_{\beta_{k+1}}}\Big[-\beta_k U\big(T_\theta^{-1}(y)\big) - \log Z_{\beta_k} + \log\big|\det J_{T_\theta^{-1}}(y)\big|\Big] \\
&= \mathbb{E}_{y\sim P_{\beta_{k+1}}}\Big[\beta_k U\big(T_\theta^{-1}(y)\big) - \log\big|\det J_{T_\theta^{-1}}(y)\big|\Big] + \text{const.}
\end{aligned}
$$

Analogously, we can derive the second KL term

$$
\begin{aligned}
D_{\mathrm{KL}}\big(P_{\beta_k} \,\|\, T_\theta^{-1} \sharp P_{\beta_{k+1}}\big) &= \int p_{\beta_k}(x)\,\log \frac{p_{\beta_k}(x)}{r_\theta(x)}\,\mathrm{d}x \\
&= \text{const.} - \mathbb{E}_{x\sim P_{\beta_k}}\Big[\log p_{\beta_{k+1}}\big(T_\theta(x)\big) + \log\big|\det J_{T_\theta}(x)\big|\Big] \\
&= \text{const.} - \mathbb{E}_{x\sim P_{\beta_k}}\Big[-\beta_{k+1} U\big(T_\theta(x)\big) - \log Z_{\beta_{k+1}} + \log\big|\det J_{T_\theta}(x)\big|\Big] \\
&= \mathbb{E}_{x\sim P_{\beta_k}}\Big[\beta_{k+1} U\big(T_\theta(x)\big) - \log\big|\det J_{T_\theta}(x)\big|\Big] + \text{const.}
\end{aligned}
$$

Summing the two directions

$$
\begin{aligned}
\mathscr{L}_{\mathrm{KL}}(\theta) &= D_{\mathrm{KL}}\big(P_{\beta_{k+1}} \,\|\, T_\theta \sharp P_{\beta_k}\big) + D_{\mathrm{KL}}\big(P_{\beta_k} \,\|\, T_\theta^{-1} \sharp P_{\beta_{k+1}}\big) \\
&= \mathbb{E}_{x\sim P_{\beta_k}}\Big[\beta_{k+1} U\big(T_\theta(x)\big) - \log\big|\det J_{T_\theta}(x)\big|\Big] + \mathbb{E}_{y\sim P_{\beta_{k+1}}}\Big[\beta_k U\big(T_\theta^{-1}(y)\big) - \log\big|\det J_{T_\theta^{-1}}(y)\big|\Big] \\
&\qquad + \text{const.}
\end{aligned}
$$

Thus, minimizing $\mathscr{L}_{\mathrm{KL}}$ is equivalent (up to θ–independent constants) to minimizing the bidirectional NLL shown above, i.e., aligning $T_\theta \sharp P_{\beta_k}$ with $P_{\beta_{k+1}}$ and $T_\theta^{-1} \sharp P_{\beta_{k+1}}$ with P_{β_k}.

Appendix B

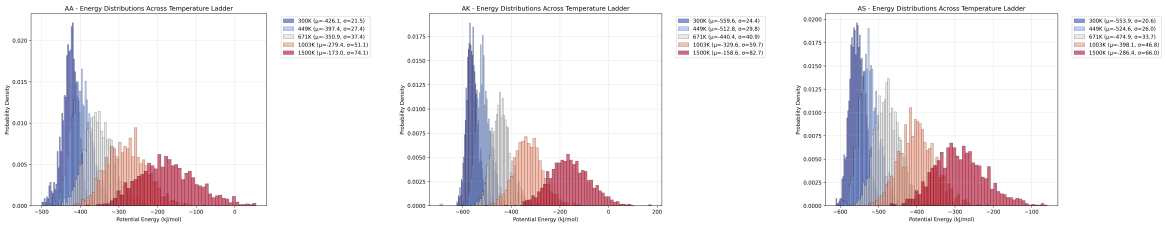
Dataset Quality

We validate the physical realism and statistical quality of our molecular datasets to ensure that subsequent flow training operates on physically meaningful data. Table B.1 reports core statistics across nine dipeptide systems at $T = 300.0\ K$. As expected, energy means scale with molecular size and chemical complexity, while standard deviations track conformational flexibility and the strength of electrostatic interactions in each system.

Table B.1 Dataset Statistics Summary

Peptide System	**N_atoms**	**Energy Mean (kJ/mol)**	**Energy Std (kJ/mol)**
AA	23	−426.6	22.6
AK	35	−559.9	24.9
AS	24	−554.6	20.6
SS	25	−637.9	21.2
KK	47	−610.9	26.3
SK	36	−622.6	34.7
SA	24	−489.4	19.8
KA	35	−530.4	21.5
KS	36	−582.1	27.3

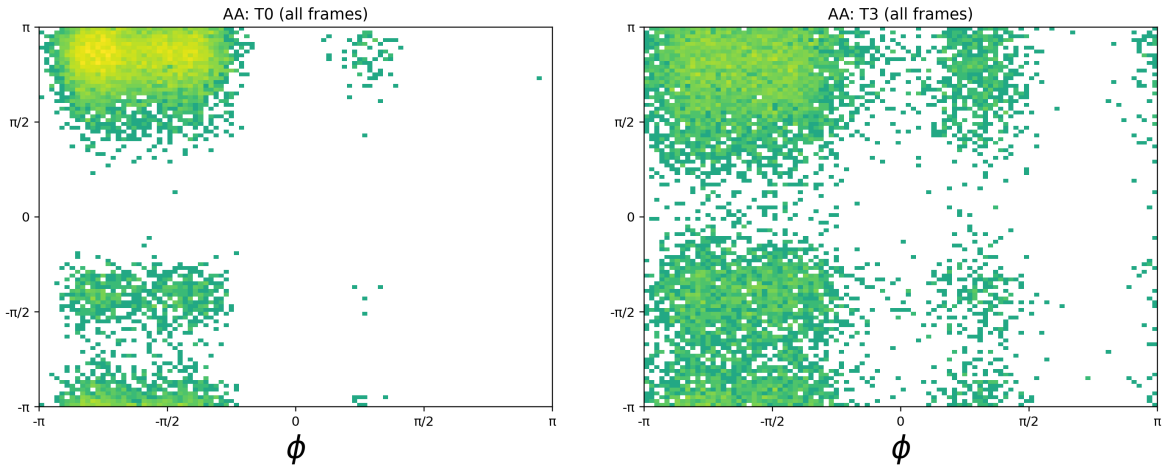
Figure B.1 shows that the energy histograms exhibit the expected temperature dependence: higher mean energies and broader spreads at elevated temperatures. The systematic broadening from $T = 300.0\ K$ to $T = 1000\ K$ indicates proper equilibration and adequate exploration of configurational space at each replica.



(a) AA Energy Distributions (b) AK Energy Distributions (c) AS Energy Distributions

Fig. B.1 Energy distribution histograms across temperature replicas.

Ramachandran analyses further confirm conformational realism across temperatures. For AA, the ϕ–ψ distributions concentrate in canonical basins at the low–temperature replica ($T = 300.0\ K$) and expand at $T = 1000.0\ K$ while retaining their underlying structure, consistent with enhanced fluctuations rather than unphysical sampling (Figure B.2).



(a) AA at $T = 300.0\ K$ (Low Temperature) (b) AA at $T = 1000.0\ K$ (High Temperature)

Fig. B.2 Ramachandran plots for AA at low and high temperature replicas.

Taken together, these checks indicate that the datasets are physically plausible, statistically well-sampled, and pose a meaningful challenge for enhanced sampling methods.