Accelerated Parallel Tempering via Neural Transports

Mila Sampling Reading Group

28/05/2025

Leo Zhang

Joint Work

Leo Zhang*
University of Oxford

Peter Potaptchik*
University of Oxford

Jiajun He*
University of Cambridge

Yuanqi Du
Cornell University

Arnaud Doucet
University of Oxford

Francisco Vargas
University of Cambridge
Xaira Therapeutics

Hai-Dang Dau
National University of
Singapore

Saifuddin Syed
University of Oxford

Original workshop paper in Frontiers in Probabilistic Inference @ICLR 2025

Published in the *Frontiers of Probabilistic Inference* workshop at ICLR 2025

GENERALISED PARALLEL TEMPERING: FLEXIBLE REPLICA EXCHANGE VIA FLOWS AND DIFFUSIONS

Leo Zhang[*†] **Peter Potaptchik**[*†] **Arnaud Doucet**[†] **Hai-Dang Dau**[‡] **Saifuddin Syed**[†]
[†]University of Oxford, [‡]National University of Singapore
[*]Equal contribution.

Neurips 2025 submission

(submission number: 18231)
(https://arxiv.org/abs/2502.10328)

Accelerated Parallel Tempering via Neural Transports

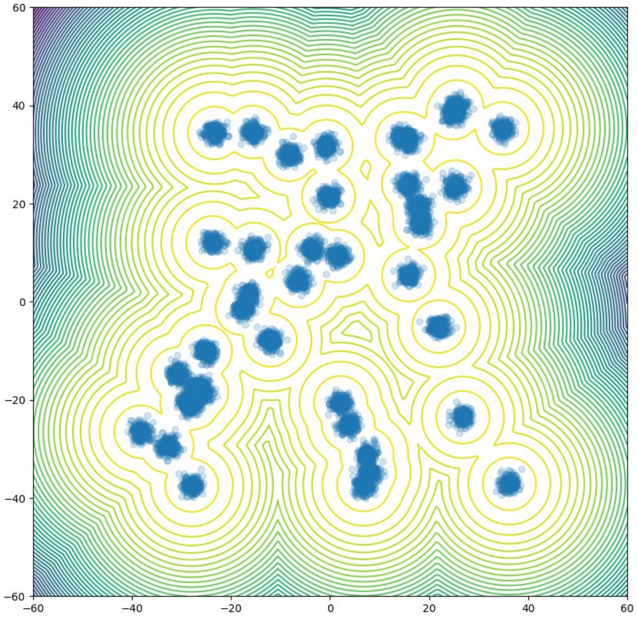
Leo Zhang[1*] **Peter Potaptchik**[1*] **Jiajun He**[2*] **Yuanqi Du**[3]
Arnaud Doucet[1] **Francisco Vargas**[2,4] **Hai-Dang Dau**[5] **Saifuddin Syed**[1]

[1]University of Oxford, [2]University of Cambridge, [3]Cornell University, [4]Xaira Therapeutics,
[5] National University of Singapore

Motivation

Sampling from probability densities is a fundamental task in:

- Machine learning
- Bayesian inference
- Molecular dynamics
- Free energy estimation

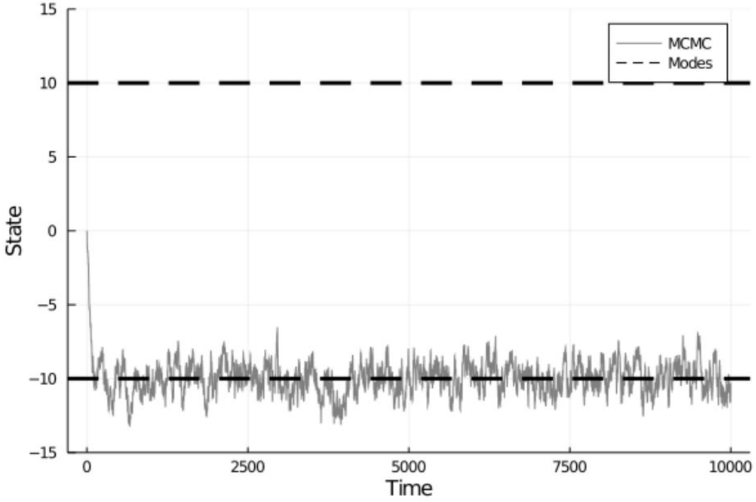


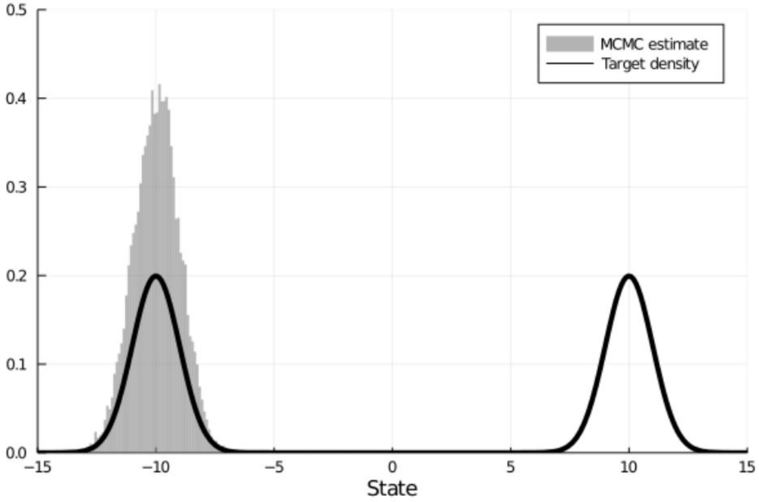
Motivation: MCMC

MCMC is a standard tool for sampling, providing mathematical guarantees

Standard MCMC methods (MALA/HMC) rely on local moves

- Suffers from mode-mixing issues leading to slow convergence

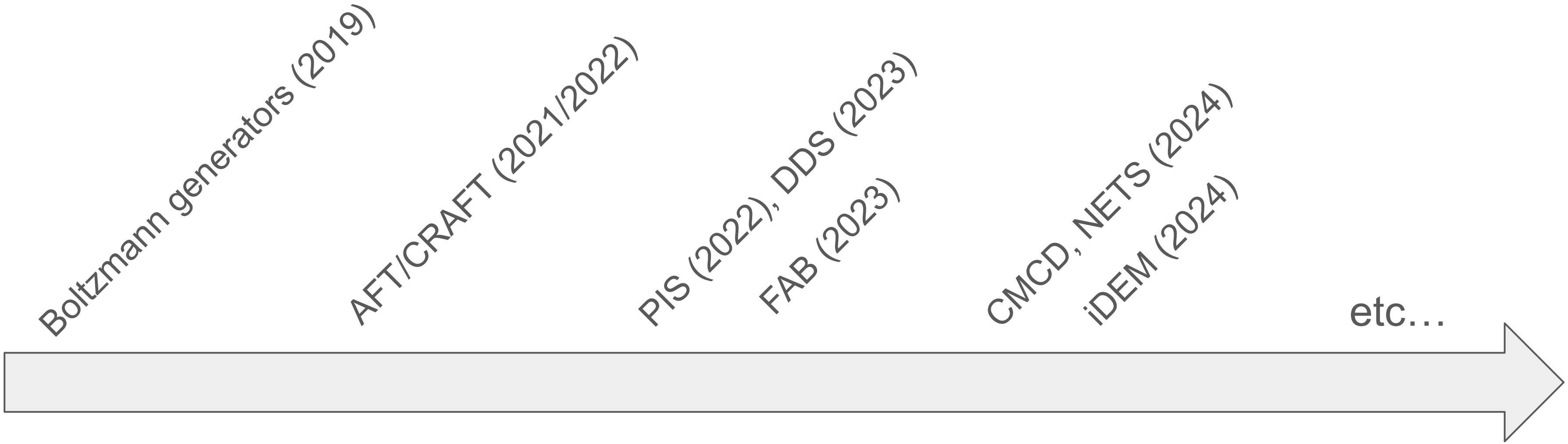




Motivation: Neural Samplers

Recent interest in leveraging advances in *generative modelling* for sampling

- We do not have access to data but can access the target density
- The cost of sampling is amortised by the trained neural network



Motivation: Neural Samplers

Despite the attractiveness of neural samplers, they suffer from foundational issues

Published at *Frontiers in Probabilistic Inference* workshop at ICLR 2025

NO TRICK, NO TREAT: PURSUITS AND CHALLENGES TOWARDS SIMULATION-FREE TRAINING OF NEURAL SAMPLERS

Jiajun He[*,1], Yuanqi Du[*,2], Francisco Vargas[1,3], Dinghuai Zhang[4], Shreyas Padhy[1], RuiKang OuYang[1], Carla Gomes[2], José Miguel Hernández-Lobato[1]

[1]University of Cambridge, [2]Cornell University, [3]Xaira Therapeutics, [4]Microsoft Research

Motivation: Neural Samplers

- Lack of mathematical guarantees
- Expensive training
- Reliance on Langevin preconditioning
- Prone to mode dropping/instability

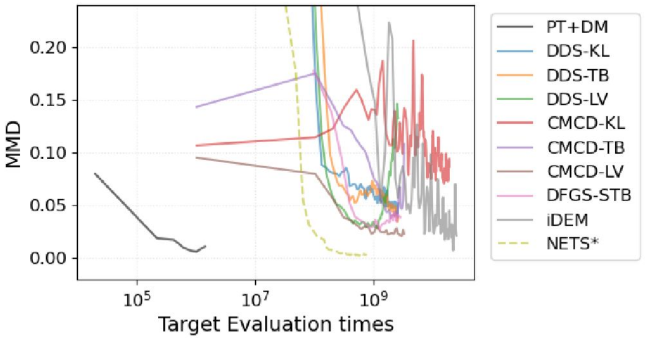


Figure 2: Sample quality vs target evaluation times for different approaches with different objectives on GMM-40 target. *NETS uses mode interpolation, which is distinct from that employed in others.

He et al. (2025) No Trick, No Treat: Pursuits and Challenges Towards Simulation-free Training of Neural Sample

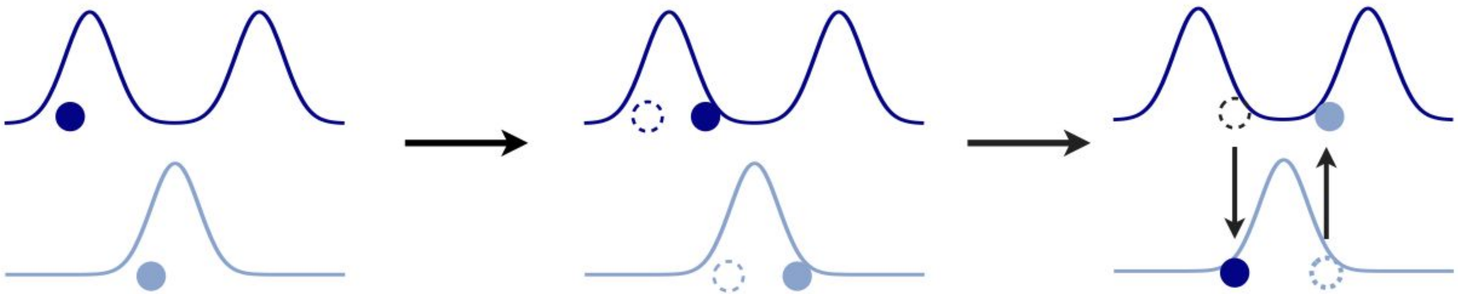
Motivation: Neural Samplers + PT

Parallel Tempering is a state-of-the-art MCMC (meta)-algorithm

Can we combine neural samples with PT?

- Shared use of *annealing*
- Precedent from SMC-based works

  (CRAFT, AFT, Particle Denoising Diffusion Sampler, Sequential Controlled Langevin Diffusions)



Taken from Saif's slides

Motivation: Neural Samplers + PT

Modern generative modelling relies on (static/dynamic) transport of measures

Ballard and Jarzynski (2009, 2012) propose incorporating "non-equilibrium switches" within PT swap moves

Replica exchange with nonequilibrium switches

Andrew J. Ballard[a] and Christopher Jarzynski[a,b,1]

[a]Institute for Physical Science and Technology, University of Maryland, College Park, MD 20742; and [b]Department of Chemistry and Biochemistry, University of Maryland, College Park, MD 20742

Edited by Bruce J. Berne, Columbia University, New York, NY, and approved May 7, 2009 (received for review January 14, 2009)

Contributions

- We formalise and generalise the framework of Ballard and Jarzynski (2009, 2012)
- We show that APT naturally provides efficient normalising constant estimators
- We provide a theoretical analysis of APT
- We illustrate the design space of APT with different neural samplers + experiments

Parallel Tempering

We consider a target distribution $\pi(x) = \exp(-U(x))/Z$

- Potential $U : \mathcal{X} \to \mathbb{R}$

We want to draw samples from π to estimate:

- Expectations $\pi[f] = \int_{\mathcal{X}} f(x)\pi(dx)$
- Normalising constants $Z = \int_{\mathcal{X}} \exp(-U(x))dx$

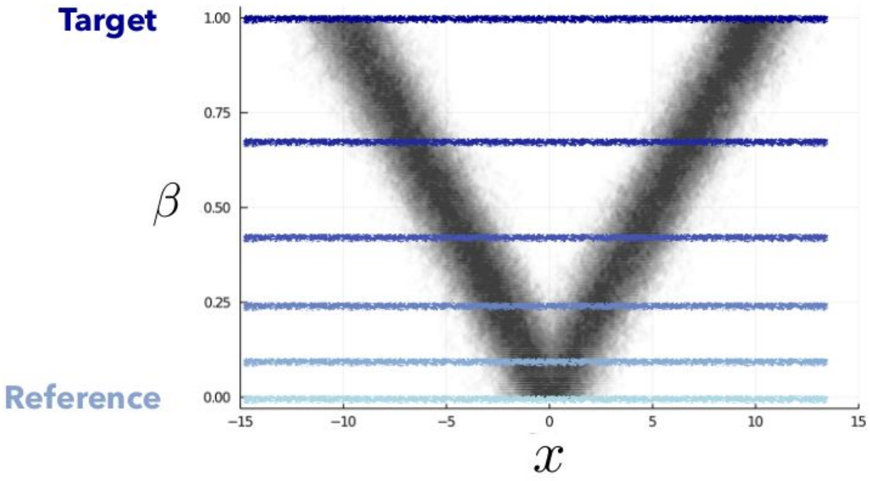
Parallel Tempering

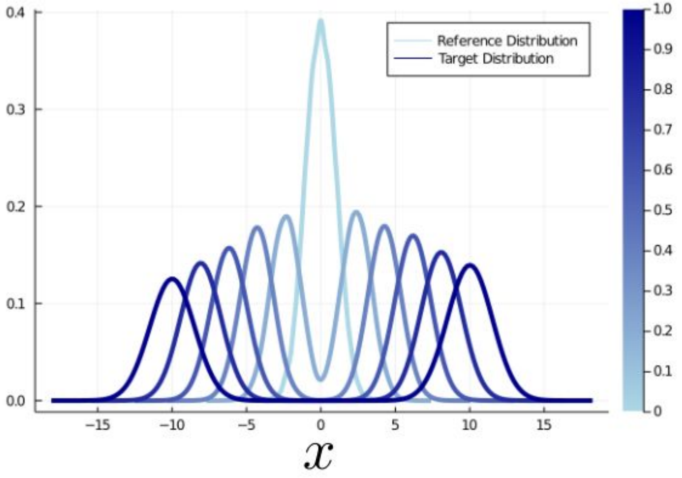
We consider an *annealing path* of distributions $\pi^0, \pi^1, \ldots, \pi^N$

- $\pi^0 = \eta$ is our reference and π^N is our target distribution
- E.g. the linear path: $\pi_\beta \propto \eta^{1-\beta}\pi^\beta$ $\quad 0 = \beta_0 < \beta_1 < \cdots < \beta_N = 1$

$$\pi^n(x) := \exp(-U^n(x))/Z_n$$

$$Z_n = \int_{\mathcal{X}} \exp(-U^n(x))dx$$





Taken from Saif's slides

Parallel Tempering

We define the *work* between π^{n-1} and π^n

$$W^n(x) = \Delta F_n - \log \frac{d\pi^n}{d\pi^{n-1}}(x) = U^n(x) - U^{n-1}(x)$$

and change in *free energy*

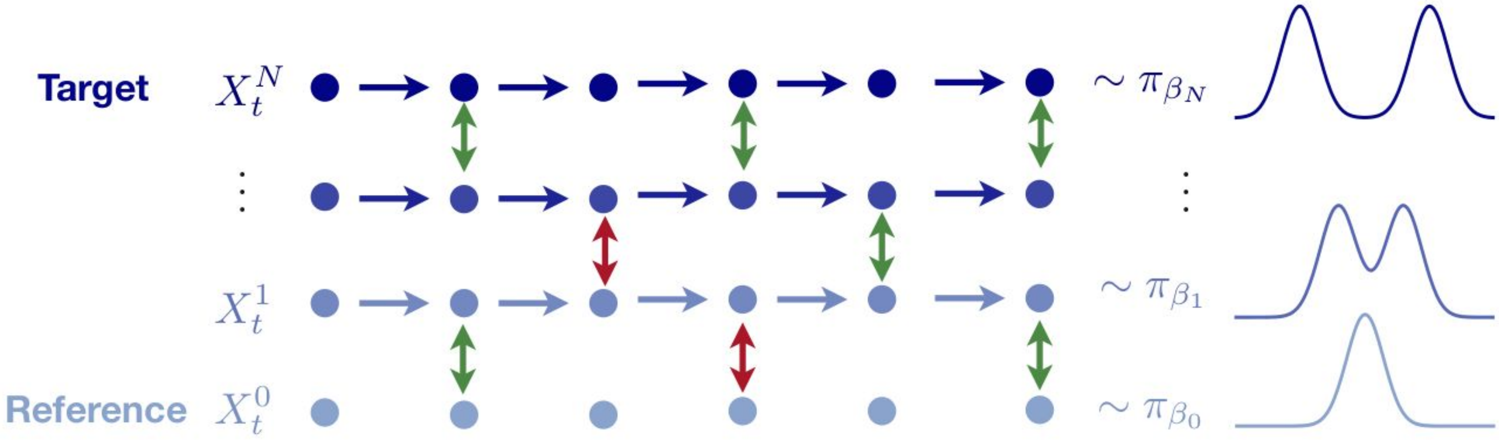
$$\Delta F_n = \log Z_{n-1} - \log Z_n$$

Parallel Tempering

We construct a Markov chain $\mathbf{X}_t = (X_t^0, \ldots, X_t^N)$ targeting $\pi^0 \otimes \ldots \otimes \pi^N$

- Local exploration: update X_t^n with a π^n-invariant kernel
- Communication: swap X_t^{n-1}, X_t^n with probability

$$\alpha^n(X_t^{n-1}, X_t^n) = \exp(\min\{0, W^n(X_t^n) - W^n(X_t^{n-1})\})$$

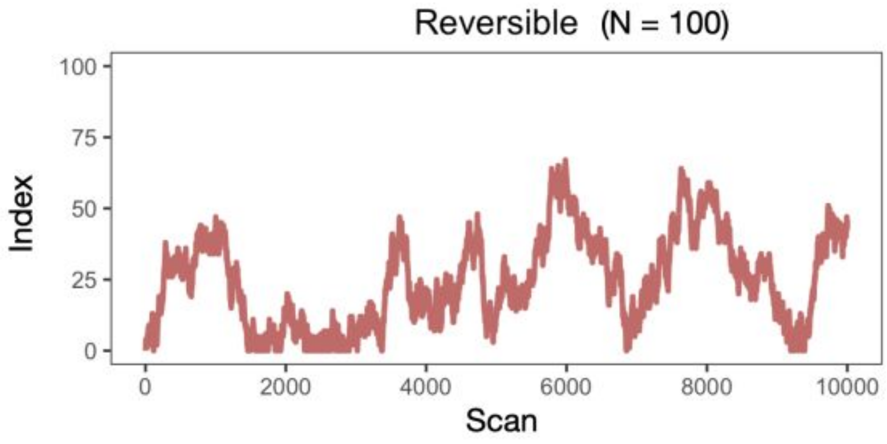


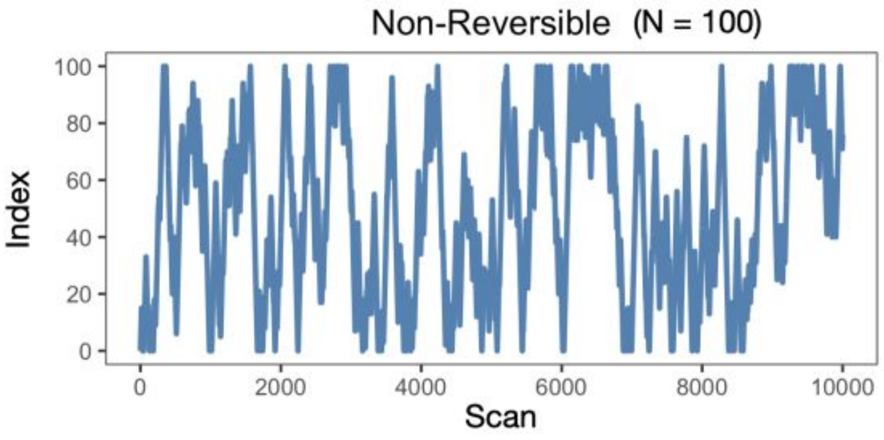
Taken from Saif's slides

Non-Reversible Parallel Tempering

For each t, we carry out swaps for the pairs (X_t^{n-1}, X_t^n) for all n in $\{n : n \equiv t (\mathrm{mod}\ 2)\}$ in parallel

- Results in non-reversible dynamics, avoiding diffusive behaviour





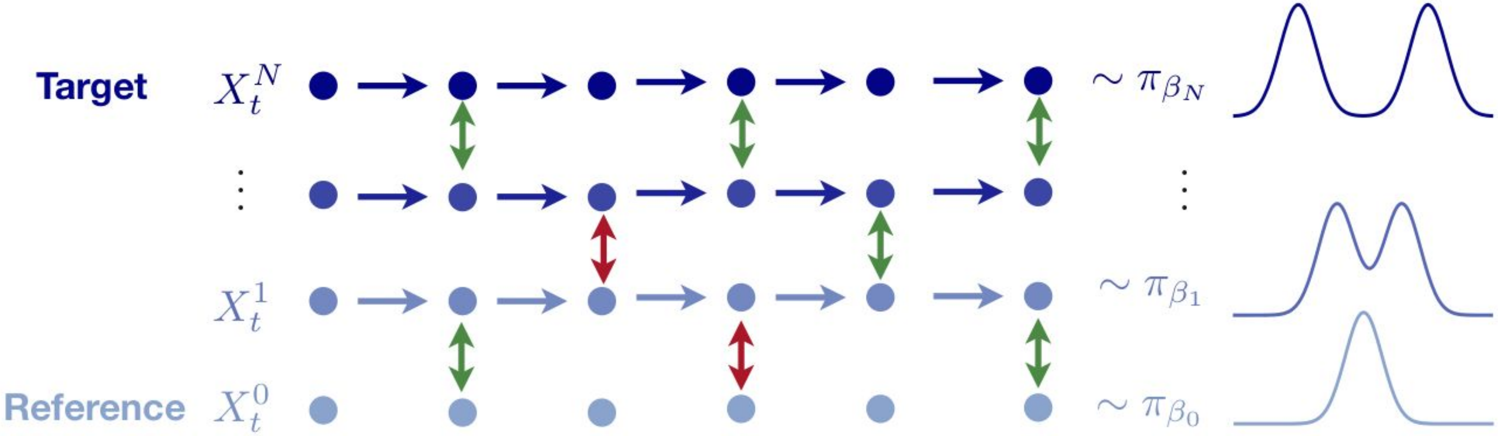
Round Trips

We measure the efficiency of PT by the *round trip rate*

This is defined in terms of the induced *machine* process tracking swaps

$$(0,1,2,3) \mapsto (1,0,3,2) \mapsto (1,0,3,2) \mapsto (0,1,3,2) \mapsto (0,3,1,2) \mapsto (3,0,2,1)$$

N=4



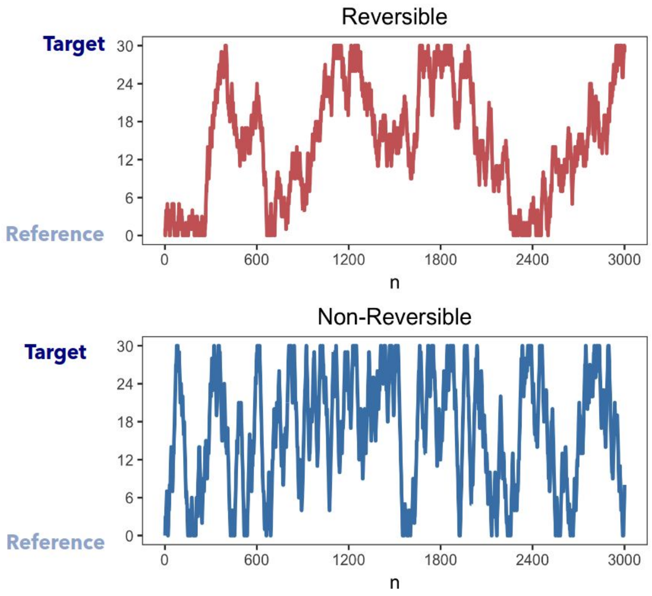
Taken from Saif's slides

Round Trips

The number of round trips is defined as the number of times a machine goes from the reference to the target and back

This serves as a good proxy for ESS but disentangles the performance of local exploration

$N = 30$



Round Trips

2

8

Rejection Rates

Under simplifying assumptions:

- Stationarity: $\mathbf{X}_t \sim \pi^0 \otimes \ldots \otimes \pi^N$
- Efficient local exploration (ELE): For $\bar{X}_t \sim \mathbf{K}^{\text{expl}}(\mathbf{X}_t, d\bar{x})$, $W^n(X_t^{n-1}), W^n(\bar{X}_t^{n-1})$ are independent and $W^n(X_t^n), W^n(\bar{X}_t^n)$ are independent

We can relate the efficiency of PT with the geometry of our path/rejection statistics

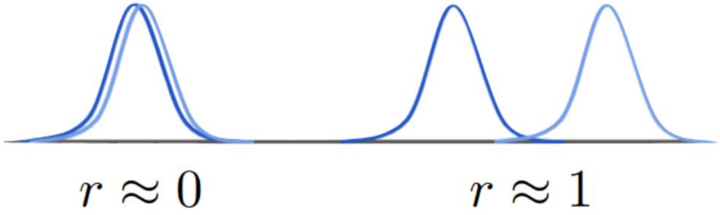
$$\tau = \left(2 + 2\sum_{n=1}^{N} \frac{r(\pi^{n-1}, \pi^n)}{1 - r(\pi^{n-1}, \pi^n)}\right)^{-1}$$

$$r(\pi^{n-1}, \pi^n) = ||\pi^{n-1} \otimes \pi^n - \pi^n \otimes \pi^{n-1}||_{\text{TV}}$$

Syed et al. (2022) Non-Reversible Parallel Tempering: a Scalable Highly Parallel MCMC Scheme

Geometry of PT

The rejection rate statistics define a divergence on our annealing path



Moreover, this provides a notion of geometry which quantifies the intrinsic difficulty of the sampling problem

Local change

$$\lambda(\beta) = \lim_{\Delta\beta \to 0} \frac{r(\pi_\beta, \pi_{\beta+\Delta\beta})}{|\Delta\beta|}$$

Global change

$$\Lambda = \int_0^1 \lambda(\beta)\mathrm{d}\beta$$

We can approximate this geometry with our rejection rate statistics

Theorem: When N is large enough, any annealing schedule satisfies,

$$r_n \approx \int_{\beta_{n-1}}^{\beta_n} \lambda(\beta)\mathrm{d}\beta, \qquad \sum_n r_n \approx \Lambda$$

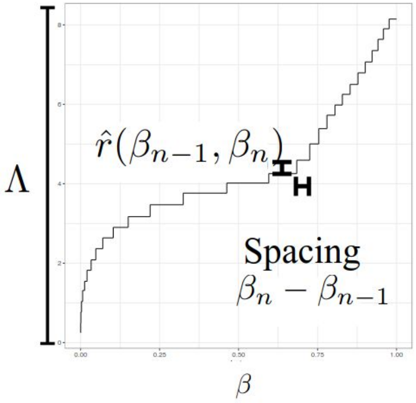
Taken from Saif's slides

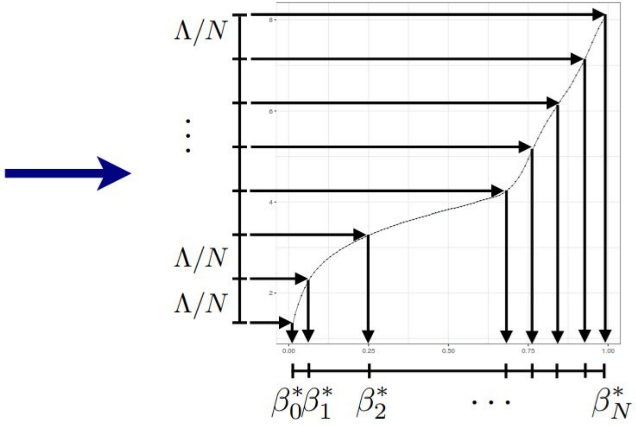
Schedule Tuning

This provides a practical algorithm for tuning the annealing schedule

- Provides state-of-the-art performance

$$\int_{\beta_{n-1}}^{\beta_n} \lambda(\beta)\mathrm{d}\beta = \frac{\Lambda}{N} \qquad r(\beta_{n-1}, \beta_n) \approx \frac{\Lambda}{N}$$





Accelerated Parallel Tempering

We’ve seen the performance of PT relies critically on the cumulative rejection rates

How can we break this barrier?

One limitation of PT is the inflexibility of the swap moves (other work has looked into learning the reference, optimising the annealing path)

- Can we take advantage of the flexibility of neural samplers to define our swap moves?

Forward and Backward Accelerators

We introduce the time-inhomogeneous Markov processes generated by the forward and backward accelerators P_k^{n-1}, Q_{k-1}^n

$$\mathbb{P}_K^{n-1}(dx_{0:K}) = \pi^{n-1}(dx_0)\prod_{k=1}^K P_k^{n-1}(x_{k-1}, dx_k) \quad \mathbb{Q}_K^n(dx_{0:K}) = \pi^n(dx_K)\prod_{k=1}^K Q_{k-1}^n(x_k,)$$

We analogously define the *work* between our accelerated paths

$$W_K^n(x_{0:K}) = \Delta F^n - \log \frac{d\mathbb{Q}_K^n}{d\mathbb{P}_K^{n-1}}(x_{0:K})$$

Non-Reversible Accelerated Parallel Tempering

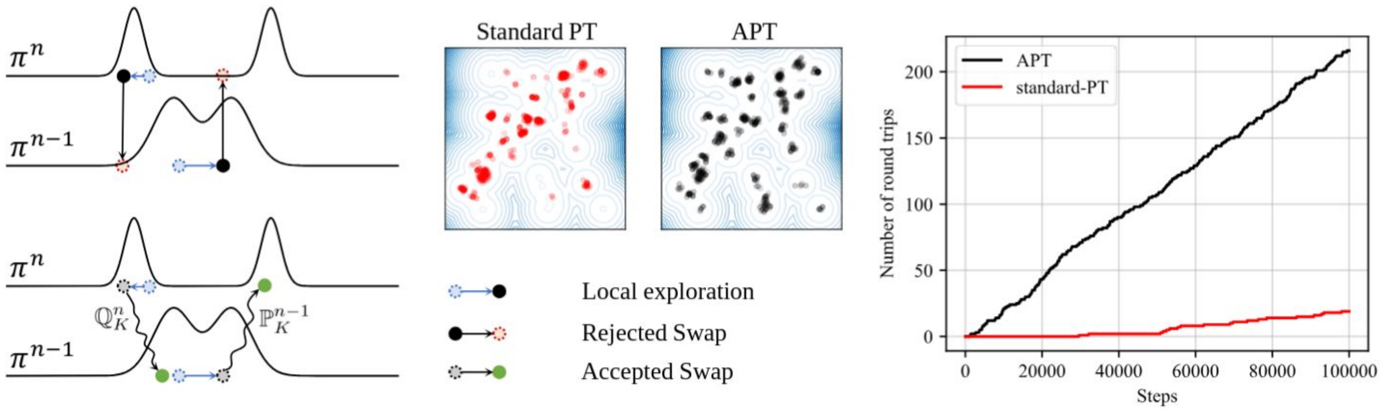


Figure 1: (Left) An illustration of the local exploration and communication step for PT vs APT. (Middle) 1,000 samples of a Gaussian mixture model target obtained using PT vs APT with a standard Gaussian reference. See Section 6.1 for more details. (Right) Round trips for PT and APT with $N = 6$ chains over $T = 100,000$ iterations of Algorithm 1.

Non-Reversible Accelerated Parallel Tempering

We define APT with the same structure as NRPT

Instead we define our swap proposal through generating the paths $\overrightarrow{X}^{n-1}_{t,0:K}$, and $\overleftarrow{X}^{n}_{t,0:K}$

$$\overrightarrow{X}^{n-1}_{t,0} = X^{n-1}_{t}, \quad \overrightarrow{X}^{n-1}_{t,k} \sim P^{n-1}_{k}(\overrightarrow{X}^{n-1}_{t,k-1}, \mathrm{d}x_k),$$
$$\overleftarrow{X}^{n}_{t,K} = X^{n}_{t}, \quad \overleftarrow{X}^{n}_{t,k-1} \sim Q^{n}_{k-1}(\overleftarrow{X}^{n}_{t,k}, \mathrm{d}x_{k-1}).$$

We then propose the new states $\overleftarrow{X}^{n}_{t,0}$ and $\overrightarrow{X}^{n-1}_{t,K}$ with probability

$$\alpha^{n}_{K}(\overrightarrow{X}^{n-1}_{t,0:K}, \overleftarrow{X}^{n}_{t,0:K}) = \exp\left(\min\left\{0, W^{n}_{K}(\overleftarrow{X}^{n}_{t,0:K}) - W^{n}_{K}(\overrightarrow{X}^{n-1}_{t,0:K})\right\}\right)$$

Expectations and Free Energy Estimation

Expectations:

- By ergodicity (Theorem 1), we have a law of large numbers result for approximating expectations

Expectations and Free Energy Estimation

Free Energy:

- We naturally have free energy perturbation/escorted Jarzynski equality estimators

$$\exp(-\Delta\overrightarrow{F}_T) := \prod_{n=1}^{N} \frac{2}{T} \sum_{n \equiv t \mod 2} \exp\left(-\overrightarrow{W}^n_{K,t}\right), \quad \exp(\Delta\overleftarrow{F}_T) := \prod_{n=1}^{N} \frac{2}{T} \sum_{n \equiv t \mod 2} \exp\left(\overleftarrow{W}^n_{K,t}\right)$$

$$\Delta\hat{F}_T = \tfrac{1}{2}(\Delta\overrightarrow{F}_T + \Delta\overleftarrow{F}_T)$$

- Moreover, we show our estimators are consistent

Proposition 1. *The estimators $\hat{\pi}^n_T[f]$ and $\Delta\hat{F}_T$ a.s. converge to $\pi^n[f]$ and ΔF respectively as $T \to \infty$. Moreover, if $\mathbb{P}^{n-1}_K = \mathbb{Q}^n_K$ for all n, then $\Delta\hat{F}_T \overset{a.s.}{=} \Delta F$.*

Algorithm 1 Accelerated Parallel Tempering

Initialise $\mathbf{X}_0 = (X_0^0, \ldots, X_0^N)$;
for $t = 1, \ldots, T$ **do**
  $\mathbf{X}_t = (X_t^0, \ldots, X_t^N), \quad X_t^n \sim K^n(X_{t-1}^n, \mathrm{d}x)$ ▷ Local exploration move
  for $n \equiv t \mod 2$ **do** ▷ Non-reversible communication
    $\overrightarrow{X}_{t,0}^{n-1}, \overleftarrow{X}_{t,K}^{n} \leftarrow X_t^{n-1}, X_t^n$ ▷ Initialise forward/backward paths
    for $k = 1, \ldots, K$ **do**
      $\overrightarrow{X}_{t,k}^{n-1} \sim P_k^{n-1}(\overrightarrow{X}_{t,k-1}^{n-1}, \mathrm{d}x)$ ▷ Accelerate forward
      $\overleftarrow{X}_{t,K-k}^{n} \sim Q_{K-k}^{n}(\overleftarrow{X}_{t,K-k+1}^{n}, \mathrm{d}x)$ ▷ Accelerate backward
    end for
    $\overrightarrow{W}_{K,t}^{n}, \overleftarrow{W}_{K,t}^{n} \leftarrow W_K^n(\overrightarrow{X}_{t,0:K}^{n-1}), W_K^n(\overleftarrow{X}_{t,0:K}^{n})$ ▷ Work of forward/backward paths
    $U \sim \mathrm{Uniform}([0,1])$
    if $\log U < \overrightarrow{W}_{K,t}^{n} - \overleftarrow{W}_{K,t}^{n}$ **then** ▷ Accelerated swap move
      $X_t^{n-1}, X_t^n \leftarrow X_{t,0}^{n}, X_{t,K}^{n-1}$
    end if
  end for
end for
Output: Return: $\mathbf{X}_1, \ldots, \mathbf{X}_T$

Analysis of APT

We demonstrate that analogous results from NRPT carry over to APT

- This allows us to carry over schedule tuning to APT
- Moreover, we show in the case of SDE bridges, scaling K improves the round trip rate

Analysis of APT

Under similar stationarity and ELE assumptions

- The rejection rates induces a divergence on our annealing path

$$r(\mathbb{P}_K^{n-1}, \mathbb{Q}_K^n) := \|\mathbb{P}_K^{n-1} \otimes \mathbb{Q}_K^n - \mathbb{Q}_K^n \otimes \mathbb{P}_K^{n-1}\|_{\mathrm{TV}}$$

- We can relate this back to the round trip rate

Proposition 2. *If Assumption 1 holds, then* $\tau = \tau(\mathbb{P}_K^{0:N-1}, \mathbb{Q}_K^{1:N})$ *where,*

$$\tau(\mathbb{P}_K^{0:N-1}, \mathbb{Q}_K^{1:N}) := \left(2 + 2\sum_{n=1}^{N} \frac{r(\mathbb{P}_K^{n-1}, \mathbb{Q}_K^n)}{1 - r(\mathbb{P}_K^{n-1}, \mathbb{Q}_K^n)}\right)^{-1}$$

Analysis of APT

We have an analogous notion of geometry for APT

- This allows us to apply the same schedule tuning algorithm from NRPT

Theorem 2. *Suppose* $\mathbb{P}_K^{\beta,\beta'}$ *and* $\mathbb{Q}_K^{\beta,\beta'}$ *are sufficiently regular and satisfy Assumptions 2–4 in Appendix B.3. As* $N \to \infty$ *if* $\max_{n \le N} |\beta_n - \beta_{n-1}| = O(N^{-1})$*, then* $\sum_{n=1}^{N} r(\mathbb{P}_K^{n-1}, \mathbb{Q}_K^{n})$ *converges to* Λ_K *and* $\tau(\mathbb{P}_K^{0:N-1}, \mathbb{Q}_K^{1:N})$ *converges to* $\bar{\tau}_K = (2 + 2\Lambda_K)^{-1}$*, where* Λ_K *equals,*

$$\Lambda_K := \int_0^1 \frac{1}{2}\mathbb{E}[|\dot{W}_K^{\beta}(\overleftarrow{X}_{0:K}^{\beta}) - \dot{W}_K^{\beta}(\overrightarrow{X}_{0:K}^{\beta})|]\mathrm{d}\beta, \quad (\overrightarrow{X}_{0:K}^{\beta} \overleftarrow{X}_{0:K}^{\beta}) \sim \mathbb{P}_K^{\beta,\beta} \otimes \mathbb{Q}_K^{\beta,\beta},$$

and $\dot{W}_K^{\beta} : \mathcal{X}^{K+1} \to \mathbb{R}$ *is the partial derivative with respect to* β' *of* $W_K^{\beta,\beta'}$ *at* $\beta' = \beta$.

Analysis of APT

We consider the case where our accelerators are given by the K-step discretisation of an underlying SDE bridging between annealing distributions

Proposition 3. *Under appropriate conditions on the drifts of the SDE (Appendix B.2), as* $K \to \infty$, $\tau(\mathbb{P}_K^{0:N-1}, \mathbb{Q}_K^{1:N})$ *converges to* $\tau(\mathbb{P}_\infty^{0:N-1}, \mathbb{Q}_\infty^{1:N})$ *and* $r(\mathbb{P}_K^{n-1}, \mathbb{Q}_K^n) \leq r(\mathbb{P}_\infty^{n-1}, \mathbb{Q}_\infty^n) + \mathcal{O}(\frac{1}{\sqrt{K}})$.

Normalising Flow Accelerated PT

Normalising flows generate samples through the push-forward of some base distribution via a differentiable, invertible mapping

- Easily computable Jacobians allow for scalable density-based training

Accelerators: $$P_1^{n-1}(x_0, \mathrm{d}x_1) = \delta_{T^n(x_0)}(\mathrm{d}x_1), \quad Q_0^n(x_1, \mathrm{d}x_0) = \delta_{(T^n)^{-1}(x_1)}(\mathrm{d}x_0)$$

Work: $$W_1^n(x_0, x_1) = U^n(x_1) - U^{n-1}(x_0) - \log|\det \nabla T^n(x_0)|, \quad x_1 = T^n(x_0)$$

The use of PT allows for flexible training, such as the symmetric KL

$$\mathcal{L}(T) = \textstyle\sum_{n=1}^{N} \mathrm{SKL}(\mathbb{P}_K^{n-1}, \mathbb{Q}_K^n)$$

Published as a conference paper at ICLR 2024

TRANSPORT MEETS VARIATIONAL INFERENCE: CONTROLLED MONTE CARLO DIFFUSIONS

Francisco Vargas*, **Shreyas Padhy***
University of Cambridge
Cambridge, UK
{fav25, sp2058}@cam.ac.uk

Denis Blessing
KIT
Karlsruhe, Germany
jl8142@kit.edu

Nikolas Nüsken*
Kings College London
London, UK
nik.nuesken@gmx.de

Controlled Monte Carlo Diffusions

Essential idea:

- Fix an annealing path between a reference distribution and target distribution
- Introduce a control term to ensure the below SDE matches the marginals of the annealing path

$$\mathrm{d}\boldsymbol{Y}_t = \left(\sigma^2 \nabla \ln \pi_t(\boldsymbol{Y}_t) + \nabla \phi_t(\boldsymbol{Y}_t)\right) \mathrm{d}t + \sigma\sqrt{2}\, \overrightarrow{\mathrm{d}}\, \boldsymbol{W}_t, \qquad \boldsymbol{Y}_0 \sim \pi_0$$

- Training via matching the discretised forward and backward path measures

$$\mathbb{E}\left[\ln \frac{\pi_0(\boldsymbol{Y}_0)}{\hat{\pi}(\boldsymbol{Y}_T)} \prod_{k=0}^{K-1} \frac{\mathcal{N}(\boldsymbol{Y}_{t_{k+1}} | \boldsymbol{Y}_{t_k} + (\sigma^2 \nabla \ln \pi_{t_k} + \nabla \phi_{t_k})(\boldsymbol{Y}_{t_k}) \Delta t_k, 2\sigma^2 \Delta t_k)}{\mathcal{N}(\boldsymbol{Y}_{t_k} | \boldsymbol{Y}_{t_{k+1}} + (\sigma^2 \nabla \ln \pi_{t_{k+1}} - \nabla \phi_{t_{k+1}})(\boldsymbol{Y}_{t_{k+1}}) \Delta t_k, 2\sigma^2 \Delta t_k)}\right]$$

Controlled Monte Carlo Diffusions APT

In our context, we can use CMCD to transport between annealing distribution for swaps

We use a linear path to bridge annealing distributions where $\phi_s^n \in [0,1]$ is monotonically increasing and $\phi_0^n = 0, \phi_1^n = 1$

$$U_s^n = (1 - \phi_s^n)U^{n-1} + \phi_s^n U^n$$

Controlled Monte Carlo Diffusions APT

We define our accelerators by uniform discretisation of the CMCD SDE

Accelerators: $P_k^{n-1}(x_{k-1}, \mathrm{d}x_k) = \mathcal{N}(x_{k-1} - (\sigma_{s_{k-1}}^n)^2 \nabla U_{s_{k-1}}^n(x_{k-1})\Delta s_k + b_{s_{k-1}}^n(x_{k-1})\Delta s_k, 2(\sigma_{s_{k-1}}^n)^2 \Delta s_k)$

$$Q_{k-1}^n(x_k, \mathrm{d}x_{k-1}) = \mathcal{N}(x_k + (\sigma_{s_k}^n)^2 \nabla U_{s_k}^n(x_k)\Delta s_k + b_{s_k}^n(x_k)\Delta s_k, 2(\sigma_{s_k}^n)^2 \Delta s_k)$$

Work: $$W_K^n(x_{0:K}) = U^n(x_K) - U^{n-1}(x_0) + \sum_{k=1}^K \log P_k^{n-1}(x_{k-1}, x_k) - \sum_{k=1}^K \log Q_{k-1}^n(x_k, x_{k-1})$$

The use of PT allows for flexible training, such as the symmetric KL

$$\mathcal{L}(T) = \textstyle\sum_{n=1}^N \mathrm{SKL}(\mathbb{P}_K^{n-1}, \mathbb{Q}_K^n)$$

Diffusion Accelerated PT

We consider a VP-SDE transporting our target distribution to a standard Gaussian

$$dY_s = -\gamma_s Y_s \mathrm{d}s + \sqrt{2\gamma_s}\mathrm{d}W_s \qquad \text{with } s \in [0,1],\, Y_0 \sim \pi$$

Time-reverse SDE: $(X_s)_{s\in[0,1]} = (Y_{1-s})_{s\in[0,1]}$

$$dX_s = [\gamma_{1-s} X_s + 2\gamma_{1-s} \nabla \log \pi_s^{\mathrm{VP}}(X_s)]\mathrm{d}s + \sqrt{2\gamma_{1-s}}\mathrm{d}W$$

We define the accelerators as the discritsation of the SDE and the form of the work is the same as CMCD-APT

We parametrise an energy-based model and iteratively train via score-matching

Comparison of Acceleration Methods

Potential calls per "machine":

- NF-APT: 2
- CMCD-APT: max(2, K+1)
- Diff-APT: max(2, K+1)
- PT: 2

Table 1: PT versus APT with different acceleration methods, targeting a 40-mode Gaussian Mixture model (GMM) target in 10 dimensions and standard Gaussian reference using $N = 6, 10, 30$ parallel chains for $T = 100,000$ iterations. For each method, we report the round trips (R), round trips per potential evaluation, denoted as compute-normalised round trips (CN-R), the number of neural network evaluations per parallel chain every iteration, and Λ estimated using $N = 30$ chains.

# Chain			$N = 6$		$N = 10$		$N = 30$	
Method	Neural Call (↓)	$\hat{\Lambda}$ (↓)	R (↑)	CN-R (↑)	R (↑)	CN-R (↑)	R (↑)	CN-R (↑)
NF-APT	1	7.198	194	97.0	1655	827.5	2441	1220.5
CMCD-APT ($K = 1$)	2	6.911	234	117.0	2126	1063.0	3264	**1632.0**
CMCD-APT ($K = 2$)	3	5.932	526	175.3	3287	**1092.7**	4767	1589.0
CMCD-APT ($K = 5$)	6	**4.822**	**1743**	**290.5**	**5525**	920.8	**6231**	1038.5
Diff-APT ($K = 1$)	2	9.025	375	187.5	1551	775.5	2820	1410.0
Diff-APT ($K = 2$)	3	7.298	748	249.3	2064	688.0	3480	1160.0
Diff-APT ($K = 5$)	6	5.795	1565	260.8	3080	513.3	4334	722.3
Diff-PT ($K = 0$)	2	8.932	204	102.0	734	367.0	1586	793.0
PT	**0**	8.346	17	8.5	681	340.5	1888	944.0

Scaling with Dimensions

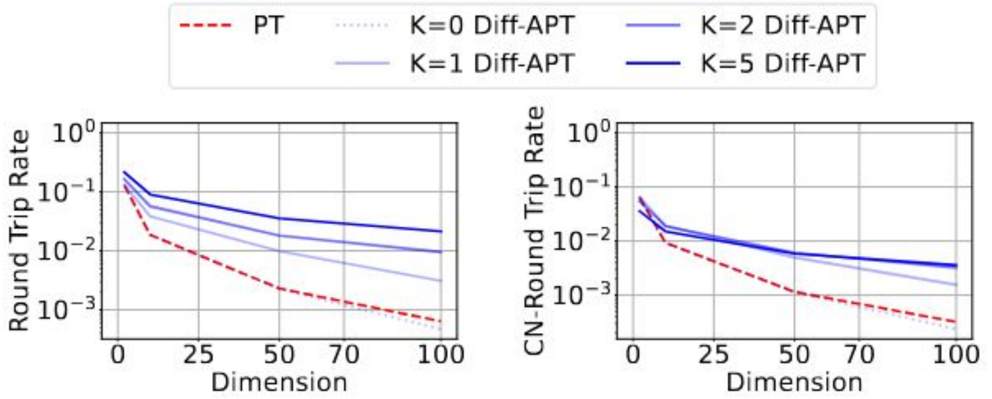


Figure 2: Round trip metrics for K-step Diff-APT ($K = 1, 2, 5$) and Diff-PT using the true diffusion path, and Linear-PT targeting GMM-d for $d = 2, 10, 50, 100$ when using 30 chains. (Left) Round trip rate against d. (Right) Compute-normalised round trip rate against d.

Free Energy Estimator

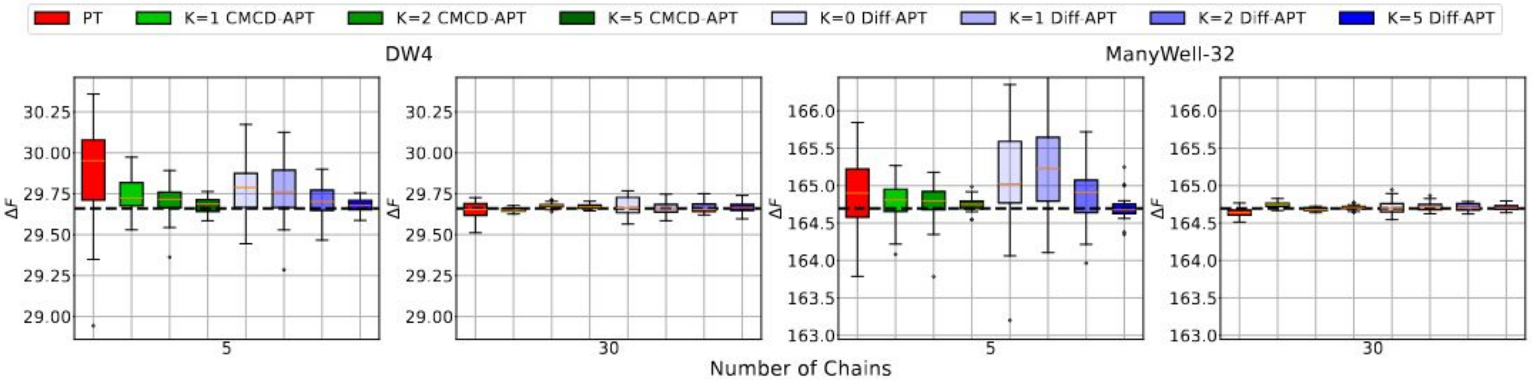


Figure 3: Estimates of ΔF for DW4 and ManyWell-32 by PT, CMCD-APT ($K = 1, 2, 5$) and Diff-APT ($K = 0, 1, 2, 5$) using 1,000 samples. Each box consists of 30 estimates. The black dashed lines denotes the reference constant $\Delta F \approx 29.660$ estimated with PT using 60 chains and 100,000 samples and $\Delta F \approx 164.696$ from Midgley et al. [2023] for ManyWell-32.

Comparing APT with Neural Samplers

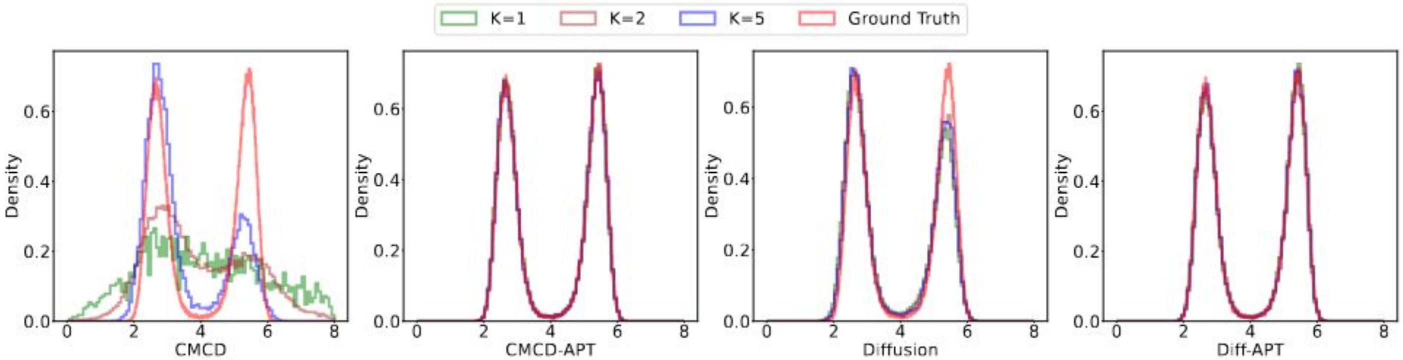


Figure 4: Interatomic distance d_{ij} of 5,000 samples by CMCD, CMCD-APT, Diffusion, Diff-APT with 30 chains, $K = 1, 2, 5$ on DW4. We take 100,000 samples by PT with 60 chains as ground truth.

Thanks

ManyWell-32

Table 2: PT versus APT with different acceleration methods, targeting ManyWell-32 in 32 dimensions and standard Gaussian reference using $N = 5, 10, 30$ parallel chains for $T = 100,000$ iterations. For each method, we report the round trips (R), round trips per potential evaluation, denoted as compute-normalised round trips (CN-R), the number of neural network evaluations per parallel chain every iteration, and Λ estimated using $N = 30$ chains.

# Chain			$N=5$		$N=10$		$N=30$	
Method	Neural Call (↓)	$\hat{\Lambda}$ (↓)	R (↑)	CN-R (↑)	R (↑)	CN-R (↑)	R (↑)	CN-R (↑)
CMCD-APT ($K=1$)	2	4.384	1154	577.0	2802	**1401.0**	4729	**2364.5**
CMCD-APT ($K=2$)	3	3.827	1587	529.0	3640	1213.3	5544	1848.0
CMCD-APT ($K=5$)	6	**3.148**	2878	479.7	4790	798.3	6678	1113.0
Diff-APT ($K=1$)	2	6.663	425	212.5	2402	1201	4398	2199
Diff-APT ($K=2$)	3	5.225	1387	462.3	4022	1340.7	5894	1964.7
Diff-APT ($K=5$)	6	3.94	**3627**	**604.5**	**5704**	950.7	**7634**	1272.3
Diff-PT ($K=0$)	2	7.423	251	125.5	1561	780.5	3440	1720
PT	**0**	5.475	550	275	1879	939.5	3733	1866.5

DW-4

Table 3: PT versus APT with different acceleration methods, targeting DW-4 in 10 dimensions and standard Gaussian reference using $N = 5, 10, 30$ parallel chains for $T = 100,000$ iterations. For each method, we report the round trips (R), round trips per potential evaluation, denoted as compute-normalised round trips (CN-R), the number of neural network evaluations per parallel chain every iteration, and Λ estimated using $N = 30$ chains.

# Chain			$N = 5$		$N = 10$		$N = 30$	
Method	Neural Call (↓)	$\hat{\Lambda}$ (↓)	R (↑)	CN-R (↑)	R (↑)	CN-R (↑)	R (↑)	CN-R (↑)
CMCD-APT ($K = 1$)	2	3.173	3020	1510.0	6407	3203.5	9456	**4728.0**
CMCD-APT ($K = 2$)	3	2.671	4239	1413.0	7549	2516.3	10538	3512.7
CMCD-APT ($K = 5$)	6	**2.107**	6971	1161.8	9808	1634.7	**12634**	2105.7
Diff-APT ($K = 1$)	2	4.565	4331	2165.5	7397	**3698.5**	7729	3864.5
Diff-APT ($K = 2$)	3	3.810	7187	**2395.7**	10176	3392	9176	3058.7
Diff-APT ($K = 5$)	6	4.358	**12456**	2076	**12740**	2123.3	8104	1350.7
Diff-PT ($K = 0$)	2	4.739	2962	1481	5862	2921	7067	3533.5
PT	**0**	4.016	2329	1164.5	5128	2564	7610	3805