

New Methods in Markov chain Monte Carlo

Max Hird, Postdoc @ University of Waterloo/ICMAT



Problem Setting

Given $\pi \in \mathcal{P}(X)$ and $f: X \rightarrow Y$ we want to calculate $\mathbb{E}_\pi[f]$

We focus on π sufficiently complex such that many key properties are unknown

Therefore we don't have $\mathbb{E}_\pi[f]$ analytically, nor can we sample independently from π to form (simple) Monte Carlo estimators

However we do want to sample $\{\theta_t\}_{t=1}^N$ approximately from $\pi^{\otimes N}$ so that

$$\frac{1}{N} \sum_{t=1}^N f(\theta_t) \approx \mathbb{E}_\pi[f]$$

Use Markov chain Monte Carlo (MCMC)

Motivating Example

Say we want to model IID data $(y_i, x_i) \in \mathbb{R}^D \times \mathbb{R}^D$ for $i \in [n]$ using the relationship

$$y = f_{\theta}(x) + \varepsilon, \varepsilon \sim \mathcal{N}(0, \mathbf{I}_D)$$

where $f_{\theta} : \mathbb{R}^D \rightarrow \mathbb{R}^D$ is a neural network (i.e. $\theta \in \mathbb{R}^d$ encodes the NN parameters) and we impose a prior $\pi_0 \in \mathcal{P}(\mathbb{R}^d)$ on θ

We want to sample from the posterior with density

$$\pi(\theta | y, x) \propto \exp \left(-\frac{1}{2} \sum_{i=1}^n \|y_i - f_{\theta}(x_i)\|^2 \right) \pi_0(\theta)$$

Q: What makes this hard? A1: The dimension of θ

Nowadays d is HUGE

Storage and computation of 1st and 2nd order information is difficult if not impossible

$$J_{f_\theta}(x) \quad \nabla^2 \log \pi(\theta | y, x) \quad \text{Cov}_\pi(\theta)$$

A2: π may be tightly concentrated on a nonlinear manifold

Overparametrisation means that certain directions in θ space will be uninformative, whereas amount of data may mean certain other directions are very informative

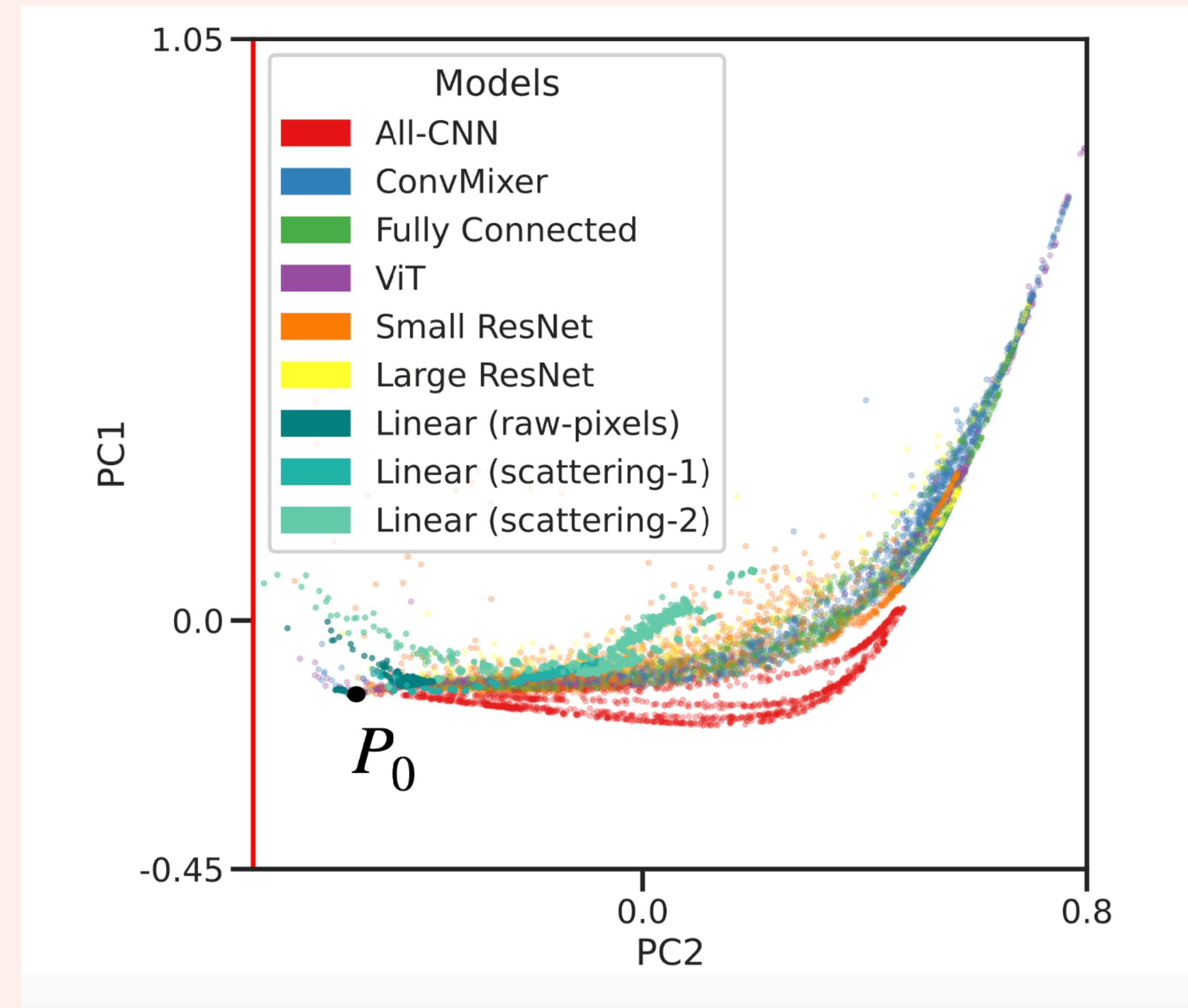
Example from [Mao et al. 2025]

Directions change as θ changes

Therefore:

π is strongly anisotropic at different points in the state space

The way in which π is anisotropic changes with the point in the state space



A3: `Genuine' Multimodality

There exist modes $\theta_1, \theta_2 \in \mathbb{R}^d$ such that $\theta_1 \neq \theta_2$ and $f_{\theta_1} \neq f_{\theta_2}$ e.g. see [Fisher and Marzouk 2025]

Cf. modes such that $f_{\theta_1} = f_{\theta_2}$ which may be found due to symmetries and are (possibly) less troublesome

Need to know *both* θ_1 and θ_2 to get a full and accurate picture of π

A4: Non-Gaussianity

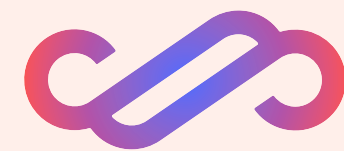
Tight concentration around a nonlinear manifold can cause gradient information to be unstable

Heavy tails cause a loss in location information

Sampling from π is hard!



We can't just use the usual MCMC methods



We propose and analyse new ones

Hold on a minute, what is MCMC?

MCMC is a way to approximately sample from π

We form a sequence of approximations $\{\mu_t\}$ such that $\mu_t \rightarrow \pi$ in some sense

In particular we form a Markov chain of samples $\{X_t\}$ such that $\{X_t\} \sim \{\mu_t\}$

So X_t is the only information we can use to sample $X_{t+1} \mid X_{1:t}$ if we want to be able to use the huge theoretical literature on Markov chains

i.e. the only information we get is the **location** of the previous state

Therefore the processes can **only move locally** using **local information** as opposed to **global information** that may be derived from $X_{1:(t-1)}$

Local processes are blessed and cursed

Moving locally allows us to build the sequence of approximations $\{\mu_t\}$ instead of attempting to sample from π in one shot

However locality hinders the vision of the Markov kernel

Markoviannity can worsen the properties of $\frac{1}{N} \sum_{t=1}^N f(\theta_t)$ as an estimator of $\mathbb{E}_\pi[f]$

So we want a process that is **still Markovian** but the **least Markovian possible**

i.e. we want the process to **mix quickly** with **low autocorrelation**

To a certain extent the **new methods** in MCMC want to **exploit global information** whilst still moving **locally**

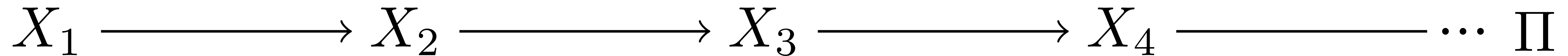
arxiv.org/abs/2604.09286

Project 1: High Dimensional Adaptive MCMC with Reduced Computational Complexity

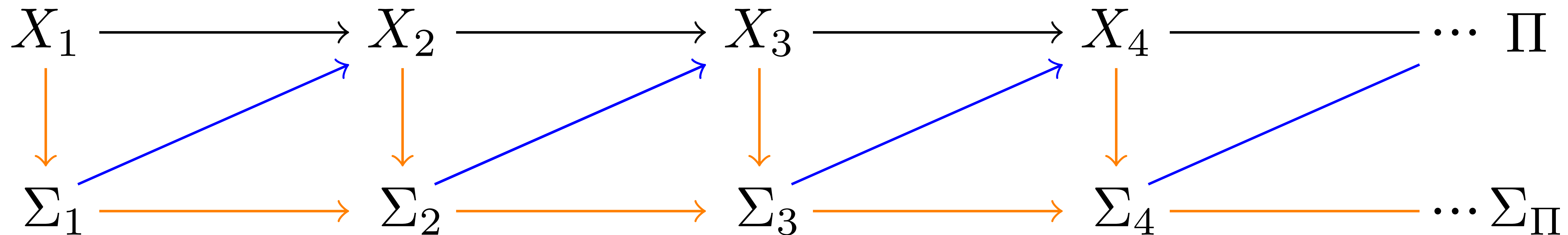


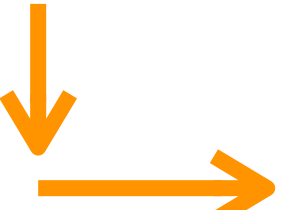
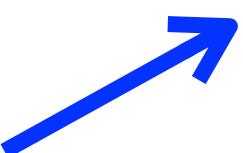
Task: sample from a target probability measure Π which is high dimensional

Vanilla Solution: Form a Π -invariant Markov chain



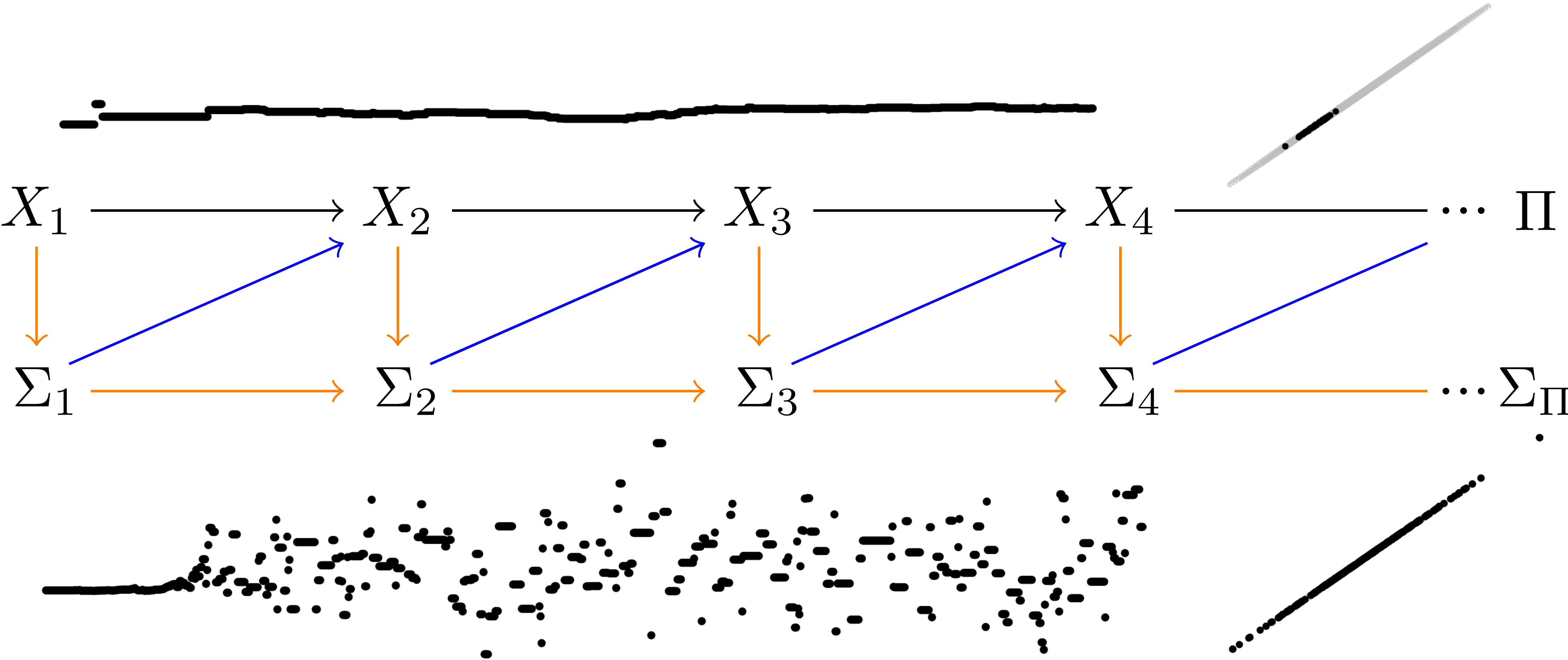
Adaptive Solution: **Learn** the covariance of Π , use to ‘**precondition**’ the Markov chain



 : Learning  : Enhancement

What's the point?

$$X_1 \longrightarrow X_2 \longrightarrow X_3 \longrightarrow X_4 \longrightarrow \cdots \Pi$$



Problem: **learning** and **preconditioning** are $O(d^2)$ (time and memory)

Solution Idea: Construct an $o(d^2)$ parametrisation $\Sigma(\theta)$ of the space of positive definite matrices, **learn** in the parameter space, reap the rewards in the **preconditioning** operations.

Challenges:

- $\Sigma(\theta)$ needs to be sufficiently representative such that using $\Sigma(\theta^*)$ makes an improvement on the vanilla solution.
- We need to **learn** and **precondition** in $o(d^2)$
- A ‘good’ parametrisation can actually be **learned**

Our approach: Eigendecomposing $\Sigma = QDQ^\top$ gives a natural $O(d + d^2)$ parameterisation. Hence we need to parametrise the space of orthogonal matrices.

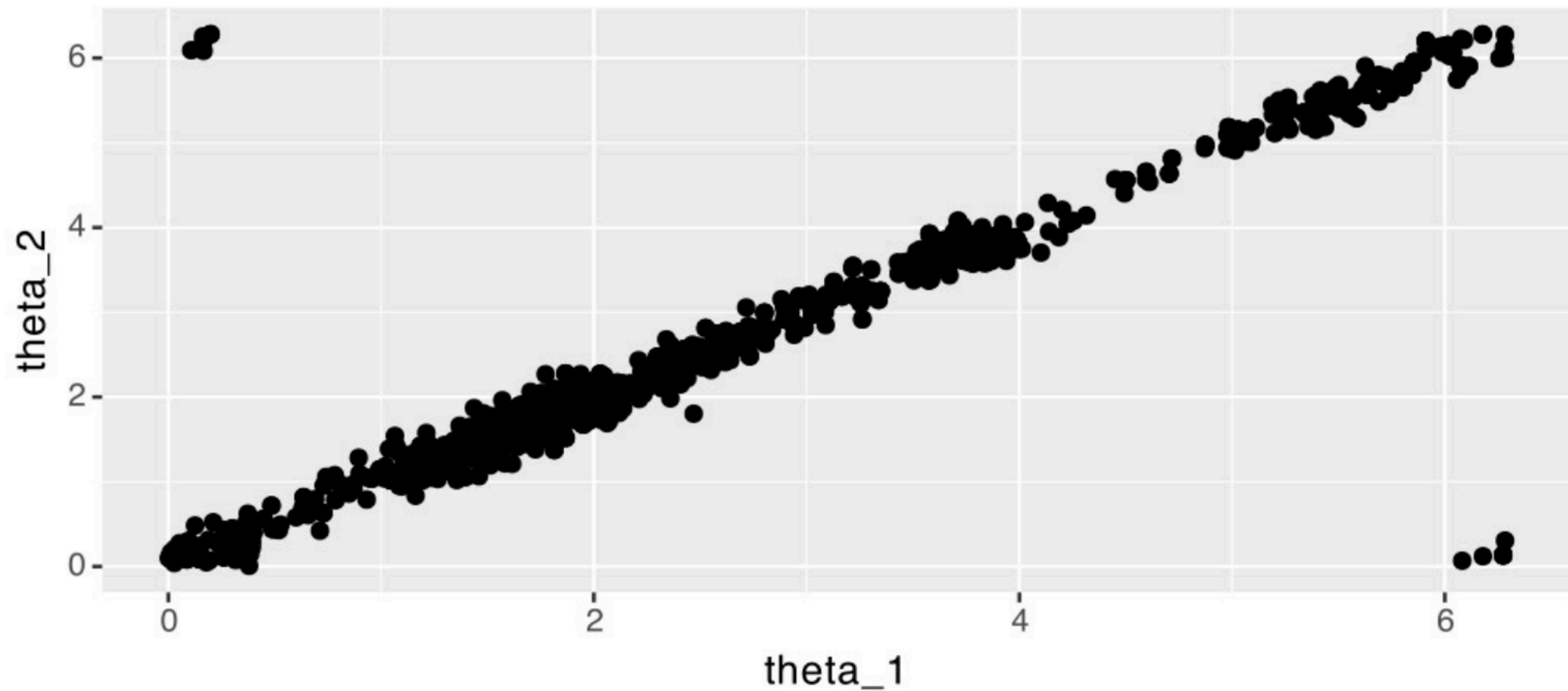
Idea: **Learn** the top k eigenvectors and eigenvalues of Σ_Π .

Parametrisation: given $V = \{v_i\}_{i=1}^k$ orthonormal vectors we construct an orthogonal $Q(V)$ whose first k columns are V using k Householder matrices.

Learning: use Oja's algorithm [Oja 1992] and Gram-Schmidt to achieve $O(k^2d)$.

Preconditioning: We know the eigendecomposition, so matrix inversion and square-root are easy. Matrix-vector products are $O(kd)$.

MALA Chain on 2 dimensional Mean Field Classical XY Target



arxiv.org/abs/2602.10714

Project 2: A Non-Asymptotic Analysis of Adaptive MCMC



Q: Can a Markov chain that learns and uses estimates of Σ_{Π} have a lower computational complexity than the same chain that does not?

Setup

Algorithm:

1. Collect $\{\tilde{X}_t\}_{t=1}^N$ using an initial chain
2. Form an estimate of Σ_Π using $\{\tilde{X}_t\}_{t=1}^N$
3. Collect $\{X_t\}_{t=1}^N$ using a re-initialised chain (i.e. with no dependency on $\{\tilde{X}_t\}_{t=1}^N$ apart from through the estimate of Σ_Π)

Termination condition:

$$W_2(\mathcal{L}(\{X_t\}_{t=1}^N), \pi^{\otimes N}) \leq \sqrt{N}\varepsilon$$

for some $\varepsilon > 0$

Assumption:

$$W_2(\mu_t, \pi) \leq \Gamma \exp(-\gamma t) W_2(\mu_0, \pi) + b$$

for all $k \in \mathbb{N}$ for some $\Gamma, \gamma > 0$ and $b \geq 0$

Theorem 3.1 (abridged)

Assume that a Markov chain $\{Y_t\}_{t=1}^N$ is collected as follows:

1. ‘Burn-in’ for t_{burn} steps to get Y_1
2. Collect another state every t_{thin} steps

Then if the underlying Markov kernel satisfies the W_2 -contraction assumption, and if

$$t_{\text{burn}} \in \Omega \left(\frac{1}{\gamma} \log \frac{1}{\varepsilon - 3\Gamma^2 b} \right) \text{ and } t_{\text{thin}} \in \Omega \left(\frac{1}{\gamma} \log \frac{1}{\varepsilon^2 - 2b^2} \right)$$

Then

$$W_2(\mathcal{L}(\{Y_t\}_{t=1}^N), \pi^{\otimes N}) \leq \sqrt{N} \varepsilon$$

Theorem 3.2 (abridged)

Let $W_2(\mathcal{L}(\{Y_t\}_{t=1}^N), \pi^{\otimes N}) \leq \sqrt{N}\varepsilon$ and form the estimate

$$\hat{\Sigma}_{\Pi} := \frac{1}{N} \sum_{t=1}^N (Y_t - \bar{Y}_t)(Y_t - \bar{Y}_t)^{\top} \text{ where } \bar{Y}_t := \frac{1}{N} \sum_{t=1}^N Y_t.$$

Assume π satisfies smoothness and strong log-convexity assumptions.

Then $\|\Sigma_{\Pi}^{-1/2} \hat{\Sigma}_{\Pi} \Sigma_{\Pi}^{-1/2} - \mathbf{I}_d\| \leq \Delta$ with probability $> 1 - \delta$

if

$$N \in \tilde{\Omega} \left(\frac{d}{\Delta} \max\{\delta^{-1}, K_1\} \right) \text{ and } \varepsilon \in \tilde{O} \left(\frac{\delta \Delta}{K_2 \sqrt{d}} \right)$$

where K_1 and K_2 depend on the anisotropy of π

Application

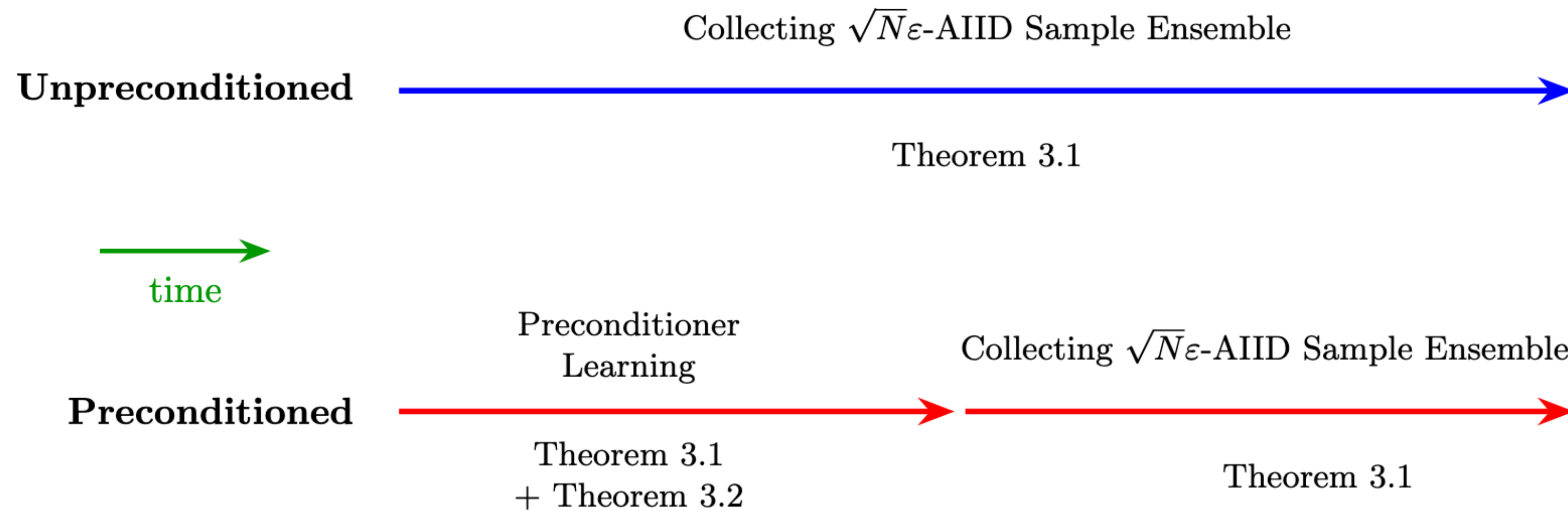


Fig. 1: Timelines of activities in the unpreconditioned Algorithm 1 (top) and the preconditioned Algorithm 2 (bottom) along with the Theorems we use to quantify their complexities.

We apply to the following MCMC algorithms:

Unadjusted Langevin (see e.g. [Dalalyan and Karagulyan 2019]), Unadjusted Kinetic Langevin (see e.g. [Cheng et al. 2018]), Unadjusted HMC of [Bou-Rabee and Marsden 2025], Proximal Sampler in [Chen et al. 2022 Section 4.2]

Extensions

For $\mu, \nu \in \mathcal{P}(\mathbb{R})$ define the Orlicz-Wasserstein distance

$$D_{\psi_2}(\mu, \nu) := \inf_{\gamma \in \Gamma(\mu, \nu)} \|X - Y\|_{\psi_2}$$

where $\|X\|_{\psi_2}$ measures the subGaussian tail decay of X

We believe that we can replicate our results in (an appropriate multivariate version of) D_{ψ_2} instead of W_2 for the algorithms mentioned before

Doing so would hopefully allow us to measure the complexity of *fully adaptive* MCMC

arxiv.org/abs/2411.11983

Project 3: Using Variational Inference to Boost MCMC



In Progress

Project 4: Riemannian MCMC with Stein Kernels



Algorithms on a manifold I

Some MCMC algorithms are conceived as discretisation of continuous time dynamics:

ULA discretises $dX_t = \nabla \log \pi(X_t)dt + \sqrt{2}dW_t$ to give

$$X_{t+1} = X_t + h \nabla \log \pi(X_t) + \sqrt{2h} \xi_t, \quad \xi_t \sim \mathcal{N}(0, \mathbf{I}_d)$$

ULA preconditioned with $M \in \text{PD}_{d \times d}$ is

$$X_{t+1} = X_t + h M \nabla \log \pi(X_t) + \sqrt{2h} M^{1/2} \xi_t, \quad \xi_t \sim \mathcal{N}(0, \mathbf{I}_d)$$

which can be viewed as a discretisation of $dX_t = M \nabla \log \pi(X_t)dt + \sqrt{2M}dW_t$

This is like the original SDE on a manifold with constant metric M

But why stop there?

Algorithms on a manifold II

Langevin dynamics on a manifold with metric $M : \mathbb{R}^d \rightarrow \text{PD}_{d \times d}$ looks like

$$dX_t = (M(X_t) \nabla \log \pi(X_t) + \text{div}(M(X_t)))dt + \sqrt{2M(X_t)}dW_t \quad [\text{Xifara et al. 2014}]$$

Various M 's have been proposed [Girolami and Calderhead 2011, Roberts and Stramer 2002, Betancourt 2019]

We are particularly interested in the case when M is a **Stein Kernel**

Gaussian integration by parts:

$$X \sim \mathcal{N}(0, \mathbf{I}_d) \iff \mathbb{E}[Xf(X)^\top] = \mathbb{E}[J_f(X)] \text{ for all } f \text{ smooth} \quad [\text{Stein 1972}]$$

A distribution $\pi \in \mathcal{P}(\mathbb{R}^d)$ has a Stein kernel $\tau_\pi : \mathbb{R}^d \rightarrow \text{PD}_{d \times d}$ when

$$\mathbb{E}[(X - \mu_\pi)f(X)^\top] = \mathbb{E}[\tau_\pi(X)J_f(X)] \text{ for all } f \text{ smooth}$$

Algorithms on a manifold III

[Cui, Tong and Zahm 2024] show that, for a particular Stein kernel τ_π

$$dX_t = (\tau_\pi(X_t) \nabla \log \pi(X_t) + \operatorname{div}(\tau_\pi(X_t)))dt + \sqrt{2\tau_\pi(X_t)}dW_t$$

decorrelates in $O(1)$ time *regardless of the details of π !*

We give equivalent conditions for π to have a Stein kernel in L^2 and identify it as a Riesz representative of the linear functional $\ell_\pi(f) := \mathbb{E}_\pi[(X - \mu_\pi)f^\top]$

Our kernel is provably different to the one in [Cui, Tong and Zahm 2024]

We provide a computational (Galerkin) method to find the kernel

Further Q's

Can one quantify convergence for a general Stein kernel?

Our computational method scales poorly with d

In Progress

Project 5: A Review of Preconditioning for MCMC

