PDMPs as
Monte Carlo algorithms models

Alexandre Bouchard-Côté
UBC

Miguel Biron

Nikola Surjanovic

Saifuddin Syed

Alexandre Bouchard-Côté

Trevor Campbell

George Deligiannidis

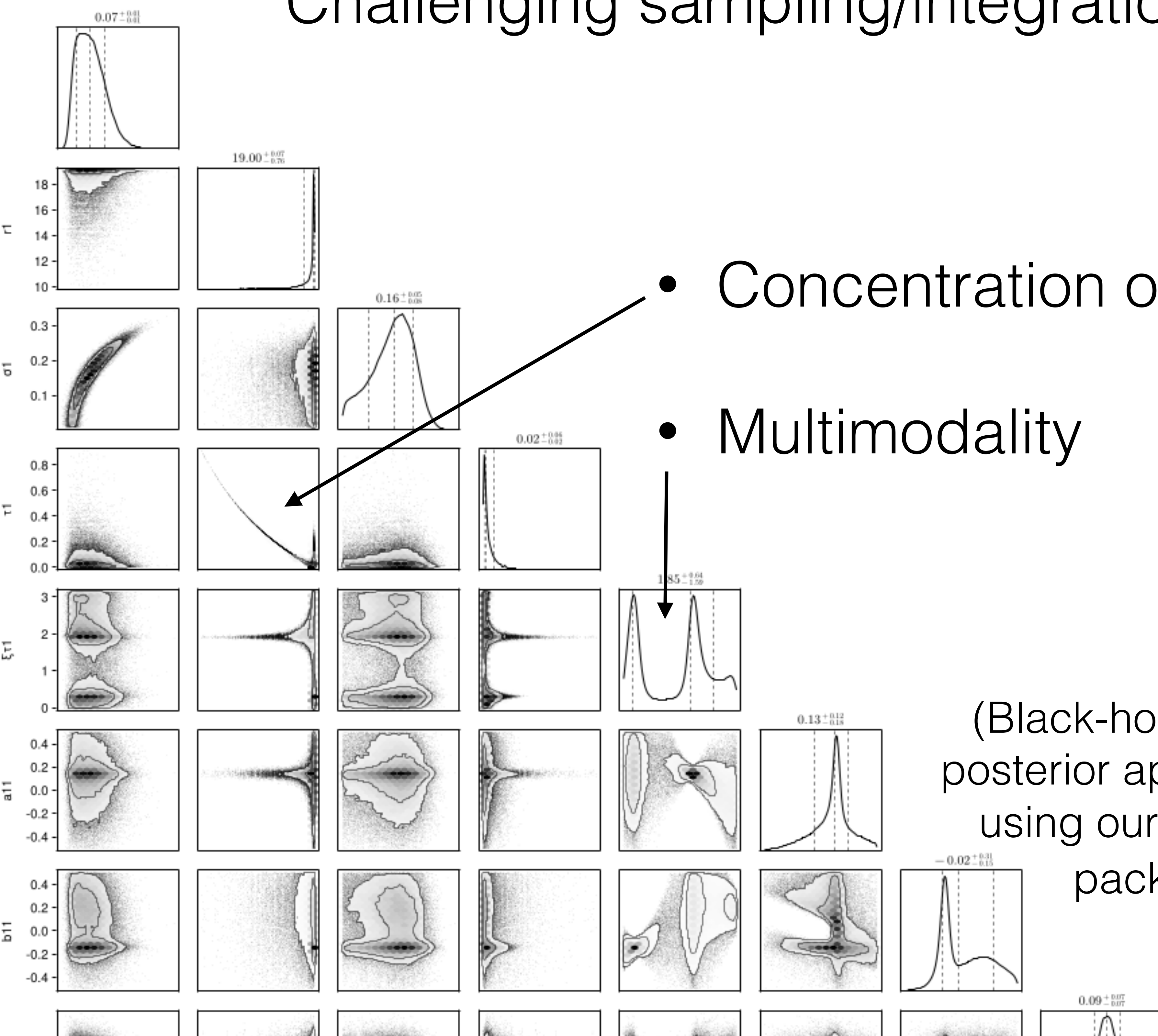
Arnaud Doucet

Outline

- Theme: models for the purpose of... ~~data analysis~~
  Monte Carlo algorithm design
  - how certain PDMPs emerge
  - importance of their event rate: the *local barriers*
- Recent work based on this “algorithm model” thinking:
  - diagnostics: *Tour Effectiveness* (TE)
  - blends of variational and MCMC methods
  - tuning non-reversible algorithms
- Open source software (`Pigeons`)

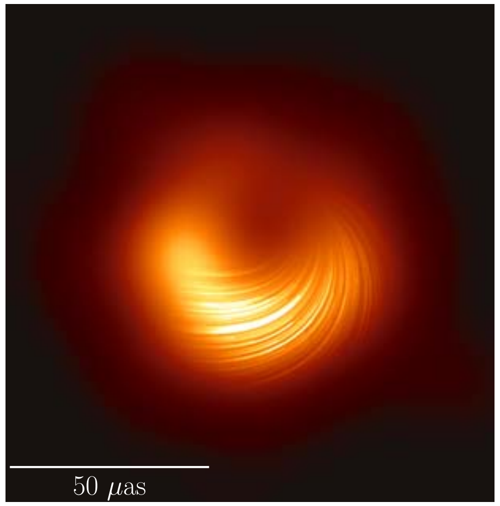
Motivation (1)

Challenging sampling/integration problems



- Concentration on sub-manifolds
- Multimodality

(Black-hole imaging posterior approximated using our `Pigeons` package)

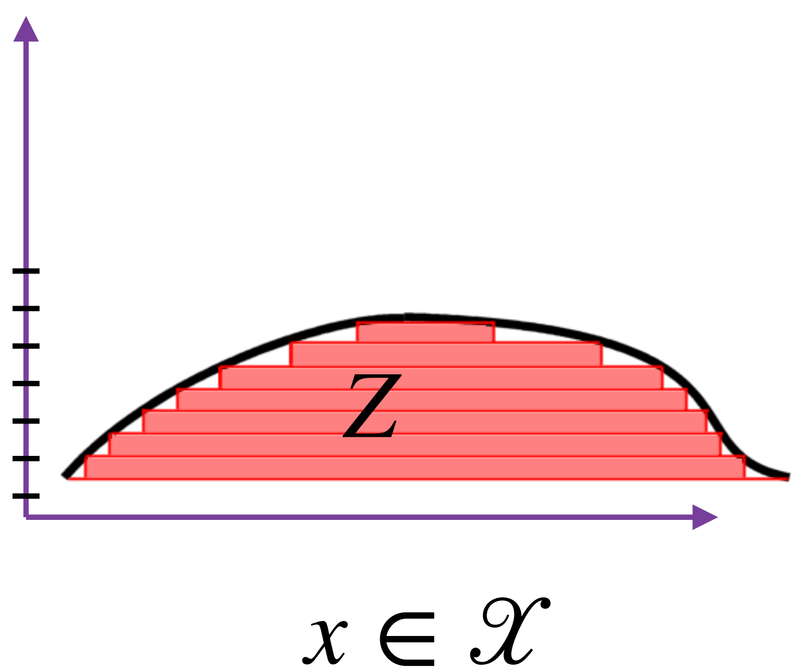


Motivation (2)

"Computational Lebesgue integration"?

$$Z = \int \exp(\ell(x))\pi_0(\mathrm{d}x)$$

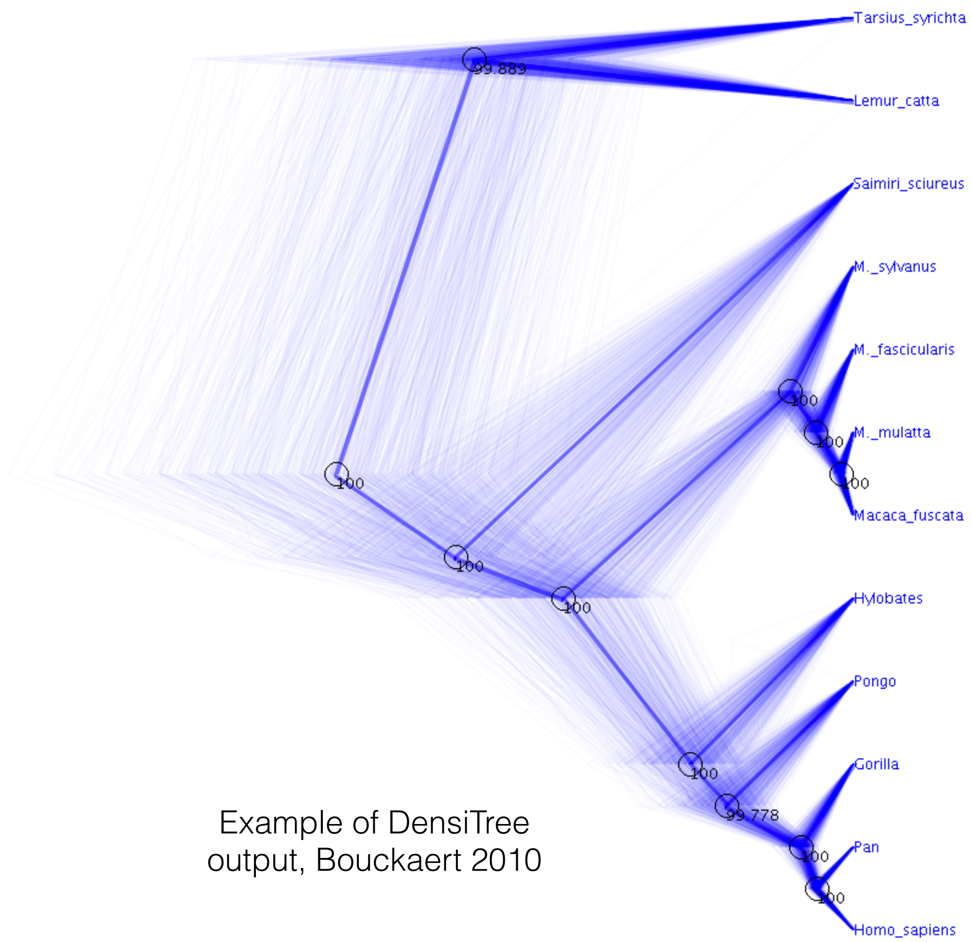
Bayes example: ℓ is the log-likelihood w.r.t. prior π_0



- Typical: assume structure on $\mathscr{X}$:
  - $\rightarrow$ to define convexity, differentiability, etc
- What can we say without structural assumptions on $\mathscr{X}$?

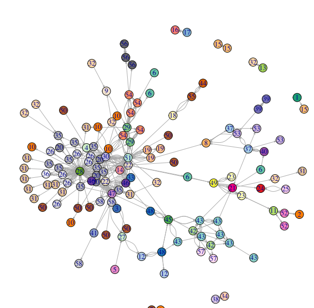
Motivation (2)

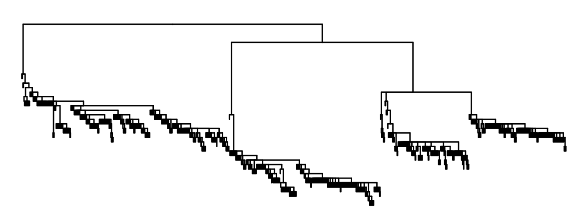
"Computational Lebesgue integration"?

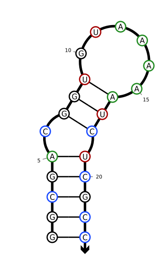


Example of DensiTree output, Bouckaert 2010

- Why care about general Lebesgue integrals?
  - statistical inference beyond vectors
  - networks, trees, molecules,...



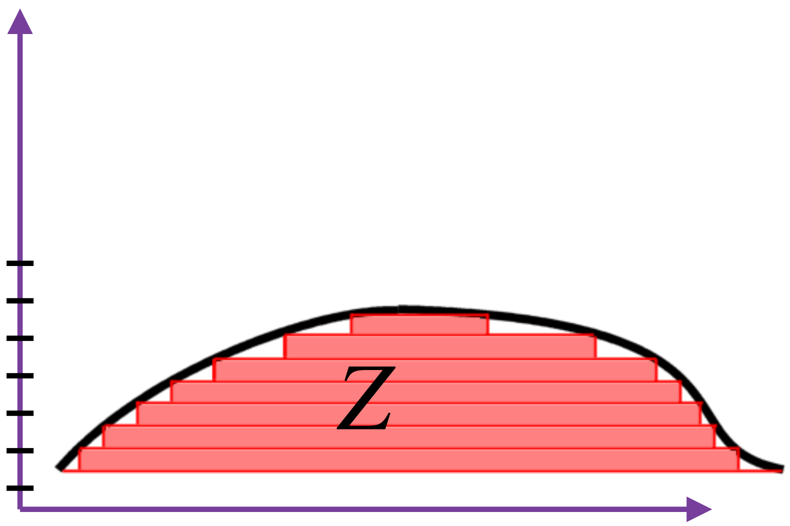




Motivation (2)

"Computational Lebesgue integration"?

$$Z = \int \exp(\ell(x))\pi_0(\mathrm{d}x)$$



$x \in \mathcal{X}$

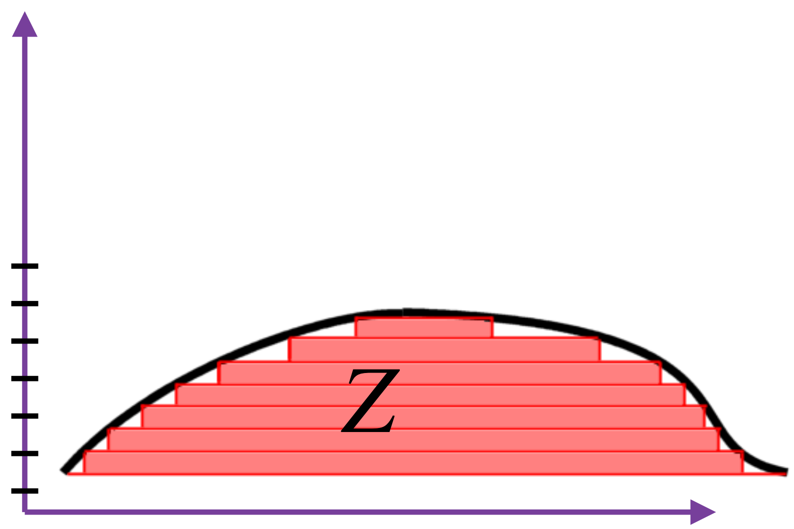
- What can we say about computing general Lebesgue integrals?
  - Too general to have algorithms
  - But *meta-algorithms* are possible...

Meta-algorithms

- **Input**: a slow-mixing "exploration" sampler,
  $X_1, X_2, \ldots$
- **Output**: a new sampler (hopefully fast-mixing?)
- **Examples**: Parallel Tempering, Simulated Tempering, ...

Meta-algorithms

$$Z = \int \exp(\ell(x))\pi_0(\mathrm{d}x)$$



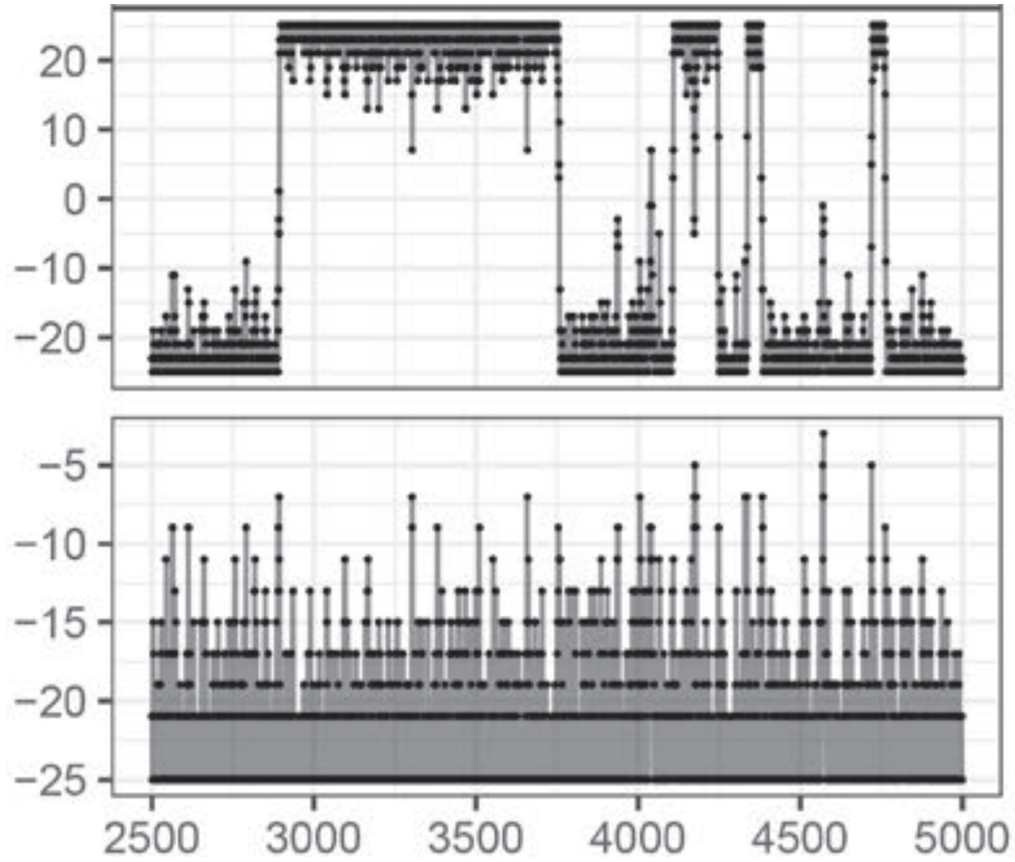
$x \in \mathscr{X}$

- Reframed question:
  - conditions where the meta-algorithm is fast-mixing...
  - ...without making structural assumptions on $\mathscr{X}$
- **Idea**: look at $Y_1, Y_2, \ldots$, where $Y_i = \ell(X_i)$

Empirical observation

Example: Ising model with Gibbs sampling

$h(X_i)$



$Y_i = \ell(X_i)$

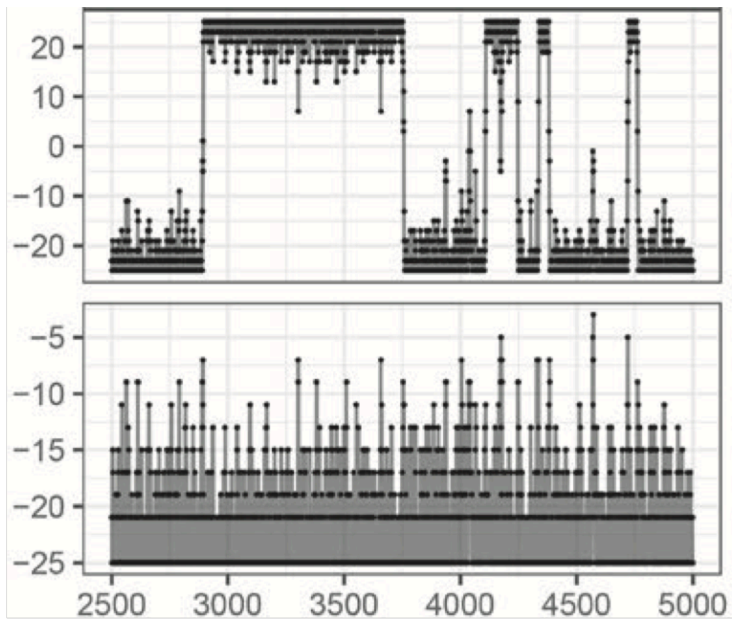
Even though X_i mixes poorly...

...the process Y_i mixes well

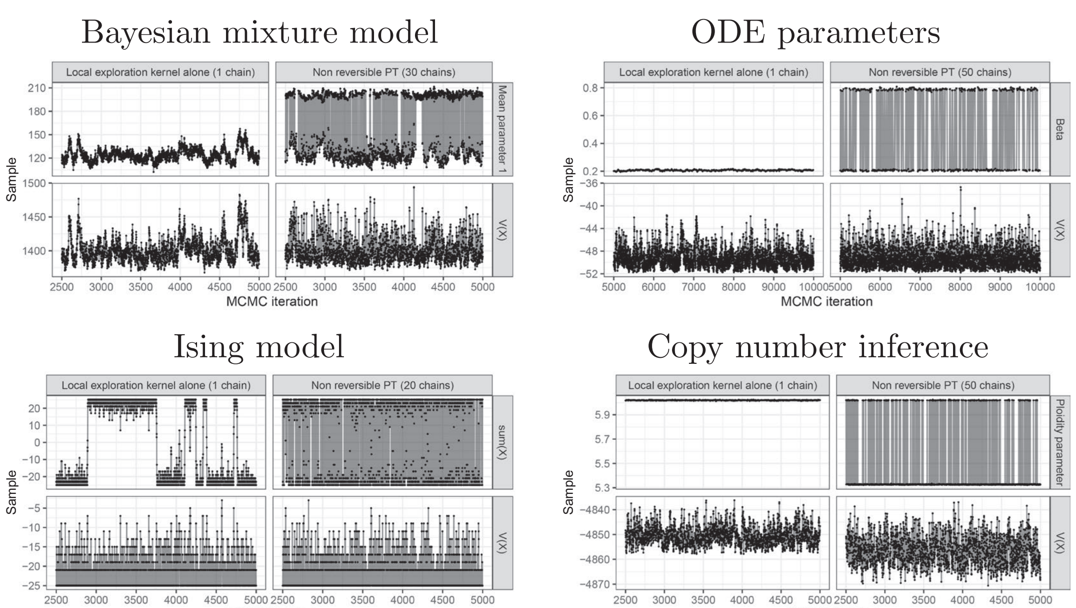
Empirical observation → model

$h(X_i)$

$Y_i = \ell(X_i)$

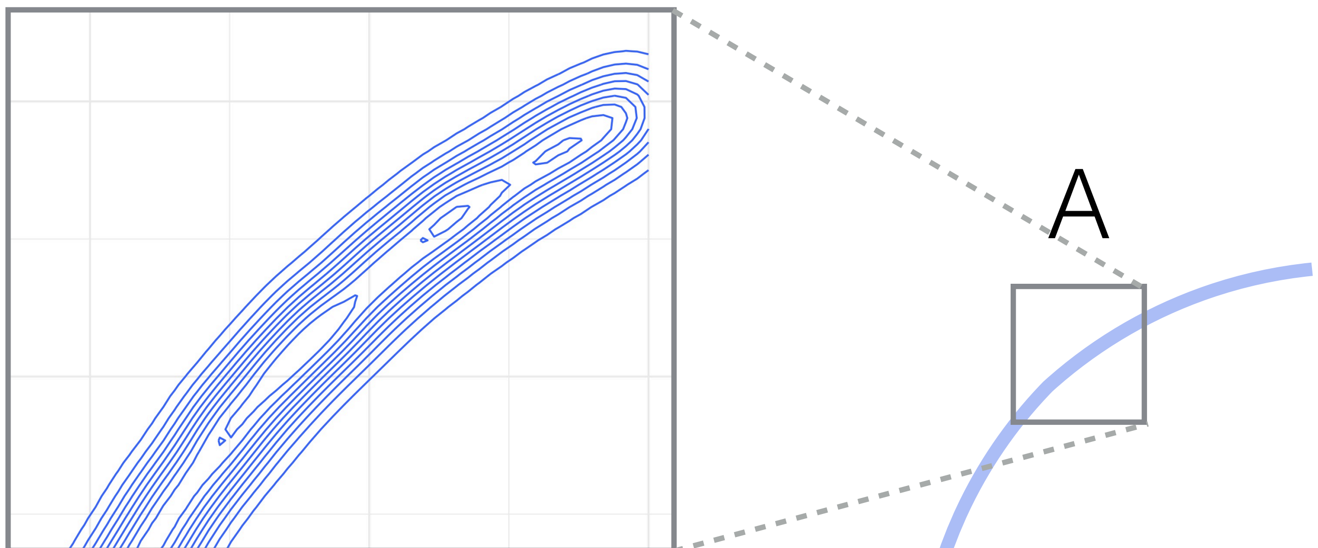


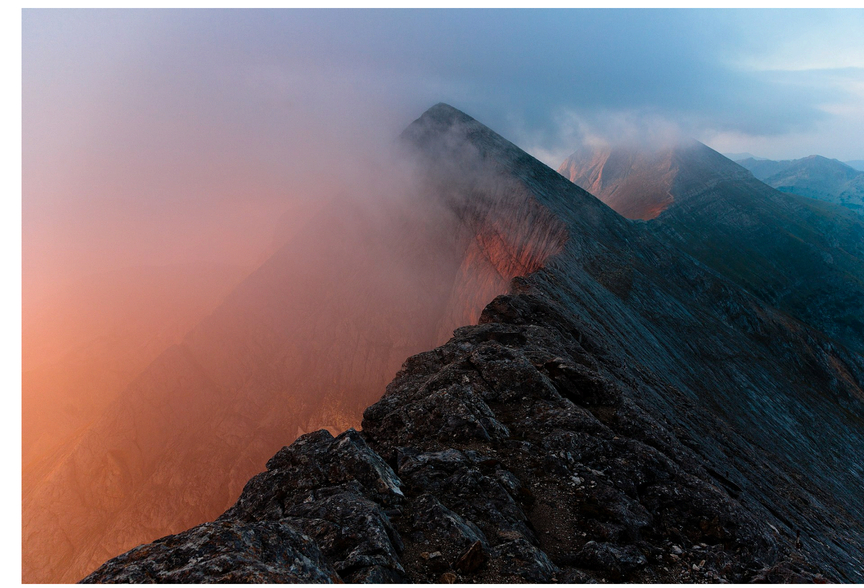
- "ELE model": $\{Y_i\}$ are assumed iid (Effective Local Exploration)
  - clearly, many problems fall outside of this regime...
  - ...but when PT mixes well, ELE often seems approximately valid

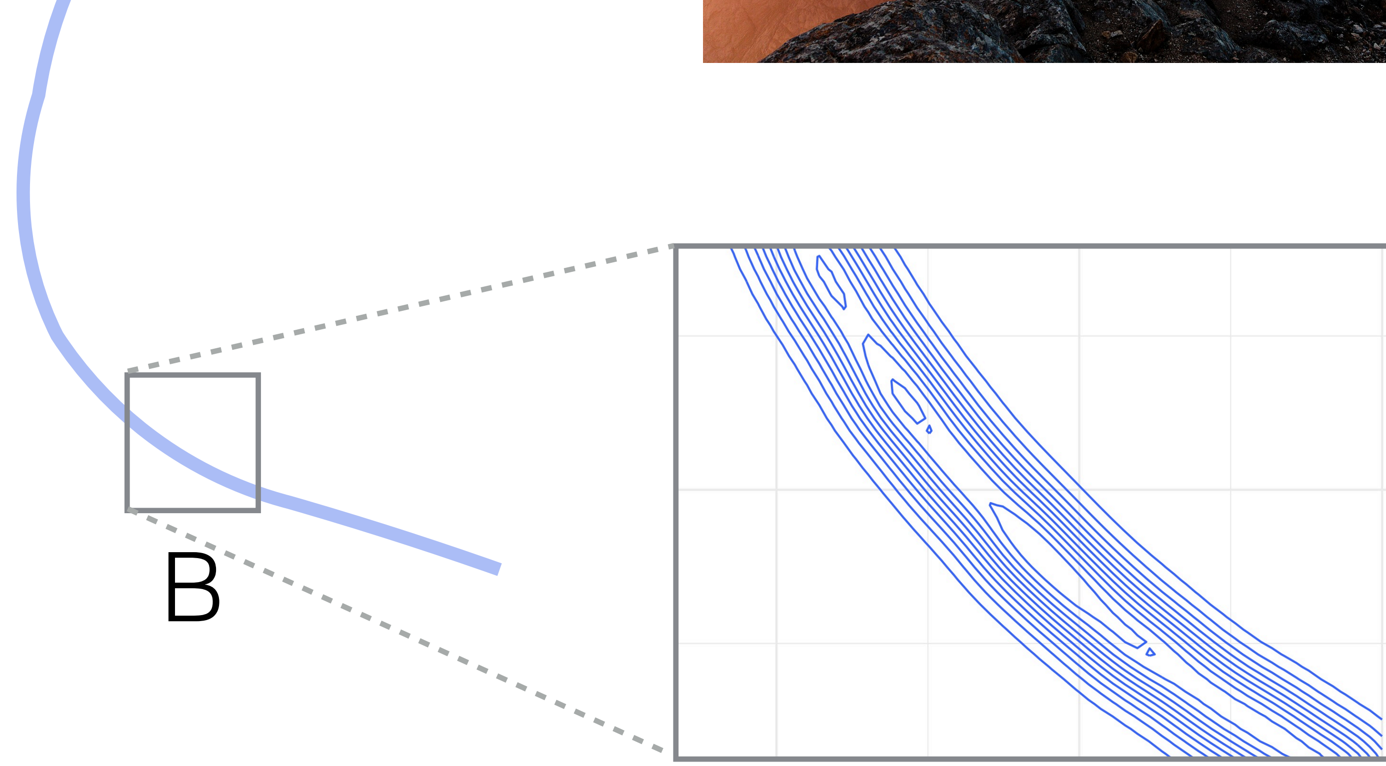


Intuition: exploring a ridge

Unidentifiable model $\Longrightarrow$ concentration on sub-manifold





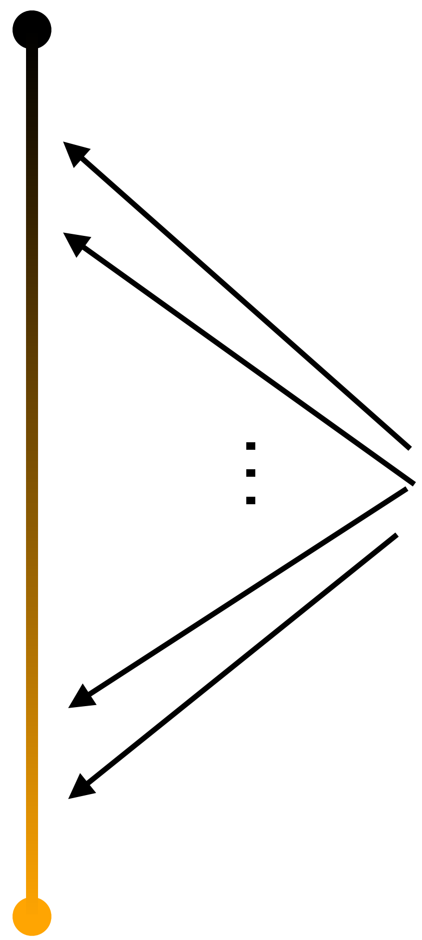


- Exploration sampler can't get from A to B...
- ...but can explore local neighbourhood, hence the "altitude distribution"

Background

Path of distributions

π_1 ← difficult



"interpolations"

Example: $$\pi_\beta(x) = \frac{1}{Z(\beta)} \exp(\beta \ell(x))\, \pi_0(x)$$

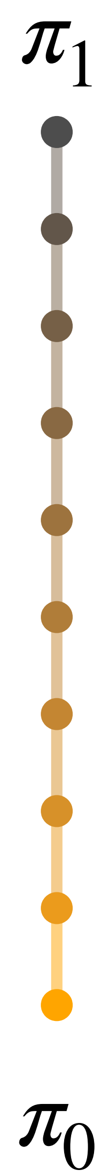
$\beta \in [0,1]$

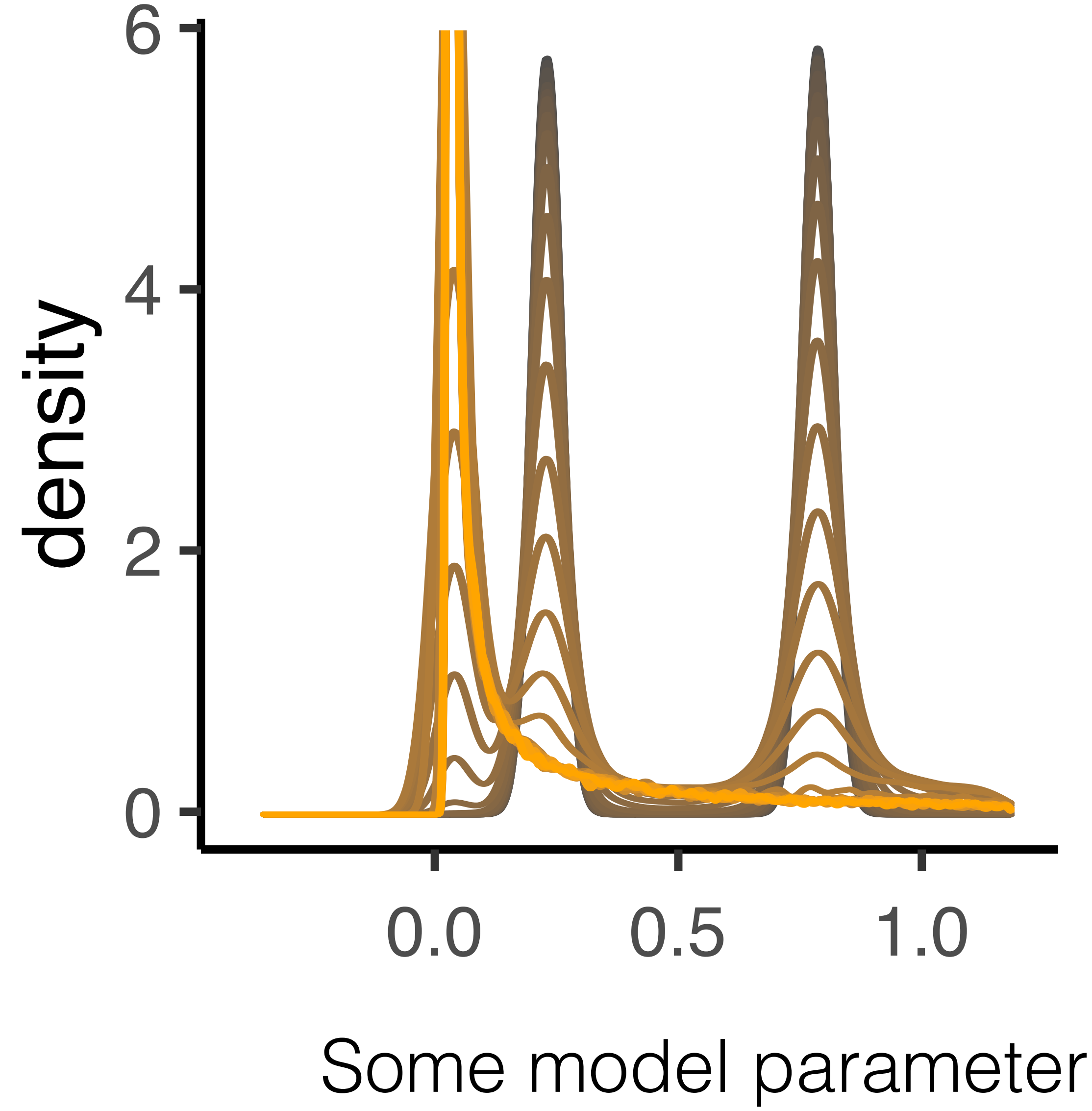
π_0 ← "Easy reference" assumption:
We can sample i.i.d. from π_0 (wlog!*)

*using a variational reference: Lefebvre et al 2009

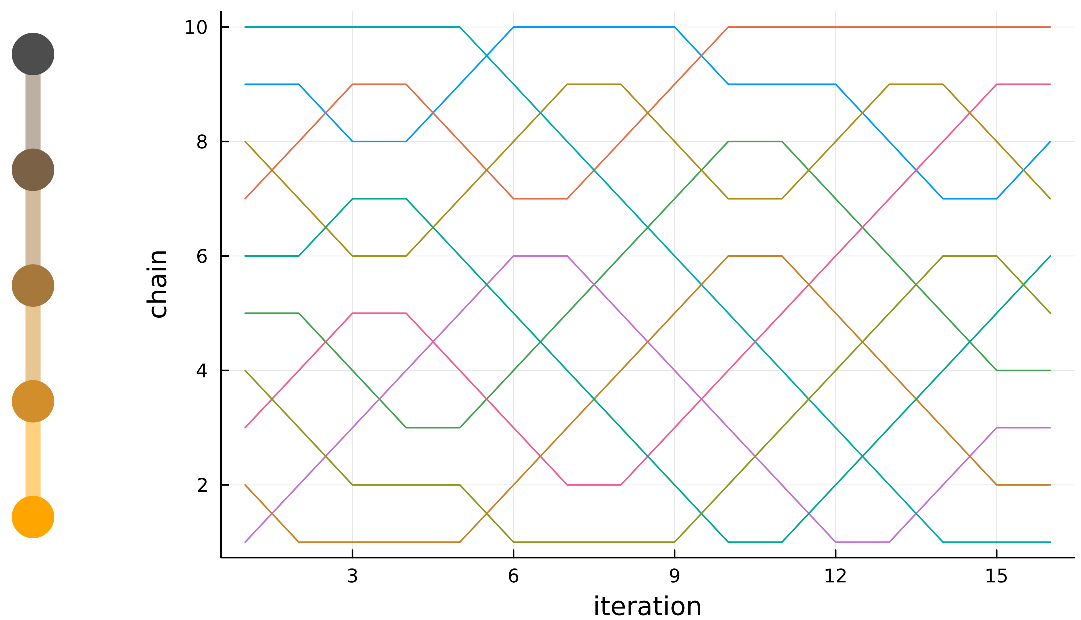
Path discretization

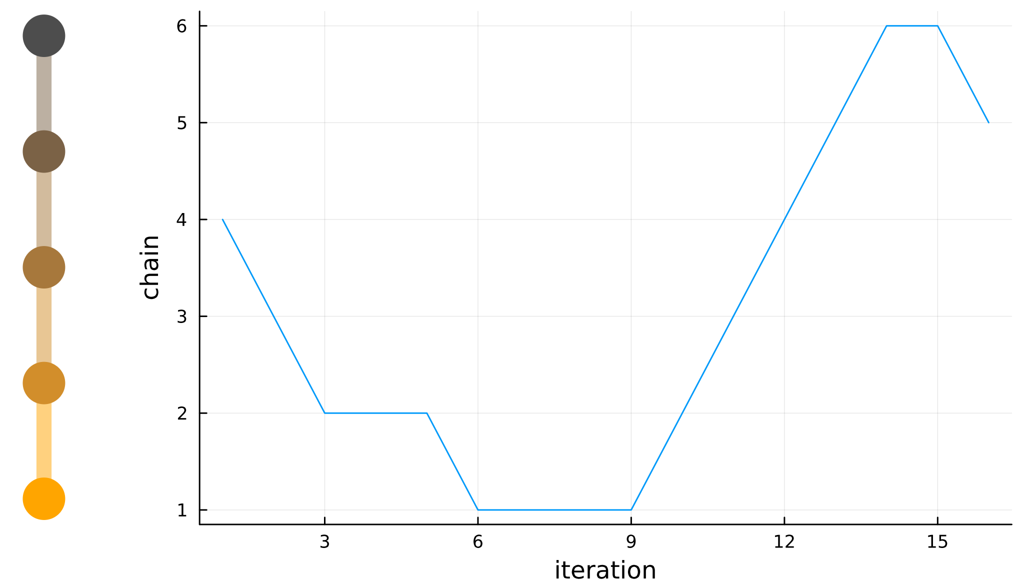
N = Number of discretization points





(**N**on-)**R**eversible annealing algorithms



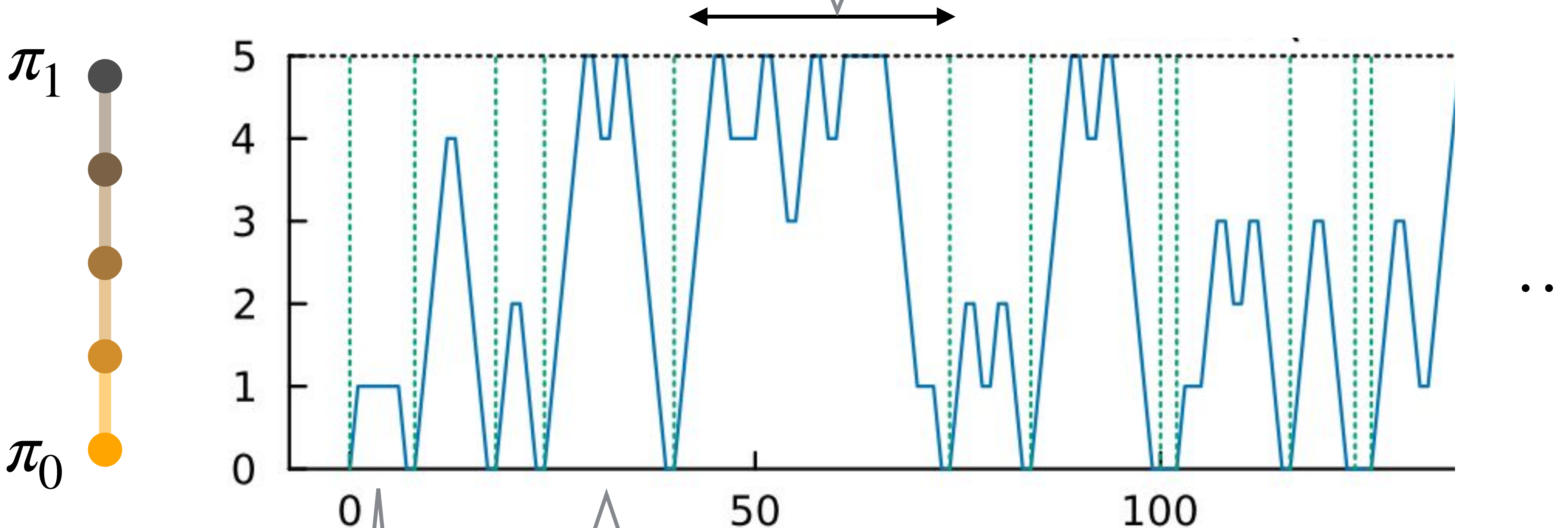


	Parallel Tempering (PT)	Simulated Tempering (ST)
R	Geyer 1991	Geyer&Thomson 1995
NR	Okabe 2001, Syed et al. 2021	Sakai&Hukushima, 2016

ST vs. PT

Only ST has a regenerative structure (through π_0)

"tour": chunk starting at $\beta = 0$ and ending at $\beta = 0$



$V_4 = 2$

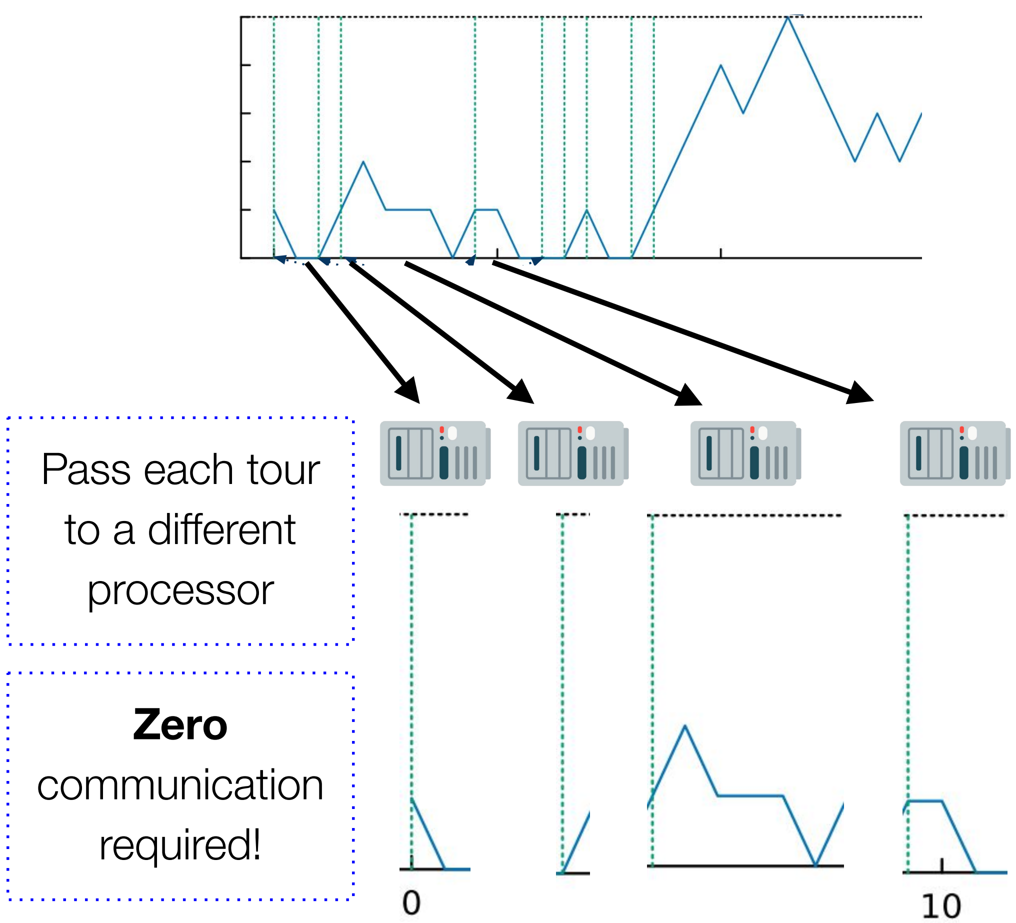
$V_1 = 0$

V_k: number of visits at $\beta = 1$ during tour k

ST vs PT

+ Easier to parallelize

+ Easier to tune



Good combo: use PT to tune ST

From ELE models to performance models

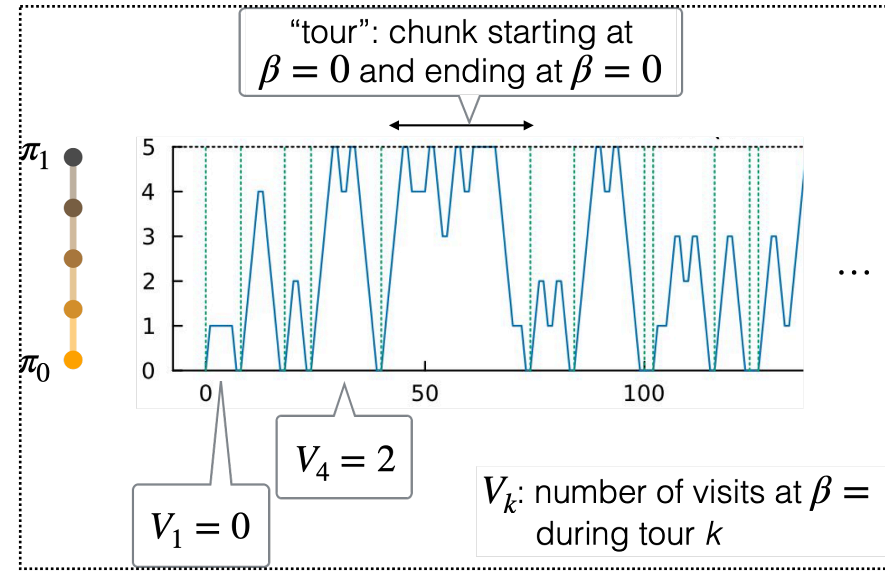
Monte Carlo standard errors

- Seek guarantees of the form:
  - "MC error is small": $\mathbb{P}(|\hat{\pi}_K(h) - \pi(h)| < \delta) \geq \alpha$

Monte Carlo estimator based on K tours

Monte Carlo standard errors

convenient bound for regenerative index processes



- For any ST algorithm, if we have:
  - an iid sampler for π_0
  - $\mathbb{E}V^2 < \infty$
  - number of tours K is large enough:

$$K \geq \frac{4}{\text{TE}} \left(\frac{z_\alpha}{\delta} \right)^2, \text{ where TE} = \frac{(\mathbb{E}V)^2}{\mathbb{E}V^2} \leftarrow \textit{Tour Effectiveness}$$

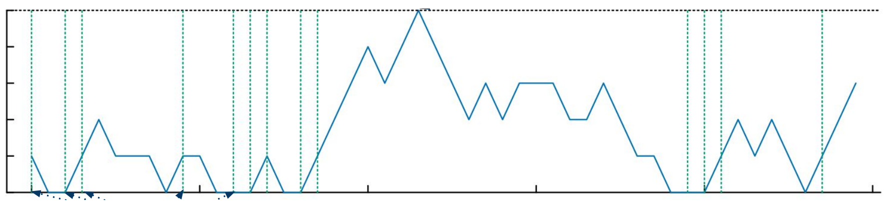
- Then MC error is small: $\mathbb{P}(|\hat{\pi}_K(h) - \pi(h)| < \delta) \geq \alpha$ for $|h| \leq 1$
- Same holds for PT, but under the ELE model

Proof ingredients: Geyer&Thomson 1995 + Kong 1992

Tour Effectiveness (TE)

$$\mathrm{TE} = \frac{(\mathbb{E}V)^2}{\mathbb{E}V^2}$$

- "Regenerative cousin" of importance sampling's ESS
- Estimation:



  - from observed tours ← use this for standard errors
  - closed form under ELE: ← for algorithm design

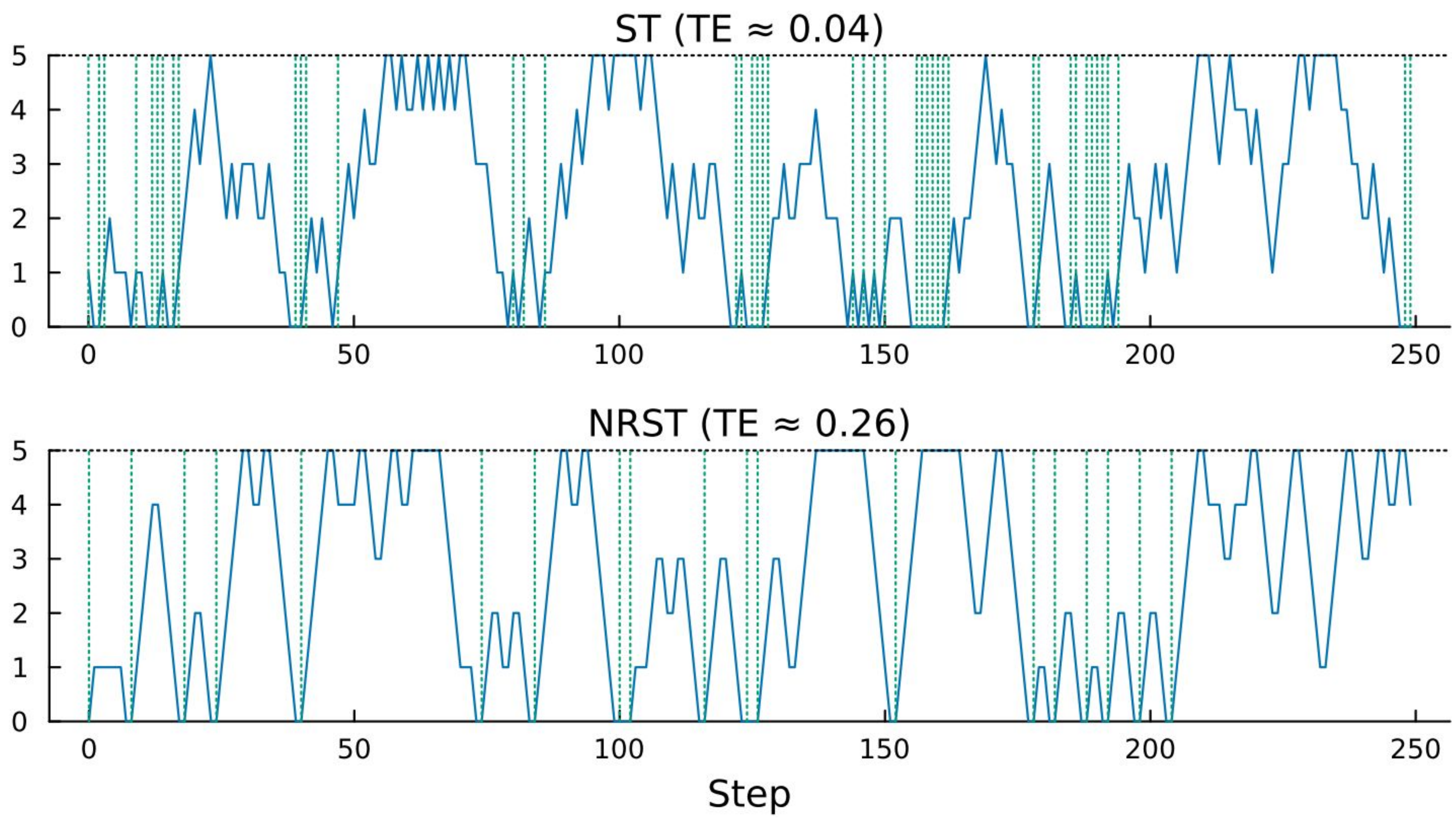
$$\mathrm{TE}_{\mathrm{NR}} = \frac{1}{1 + 2\sum_i \frac{r_i}{1 - r_i}} \qquad \mathrm{TE}_{\mathrm{R}} = \frac{4}{11}\left(\frac{1}{2N - 1 + 2\sum_i \frac{r_i}{1 - r_i}}\right)$$

Swap rejection rate between chain *i*, *i*+1

Biron et al. 2023+

Non-asymptotic comparison of (R/NR)(PT/ST)

Under ELE: $\mathrm{TE}_{\mathrm{NR}} > \mathrm{TE}_{\mathrm{R}}$ for all $N > 2$

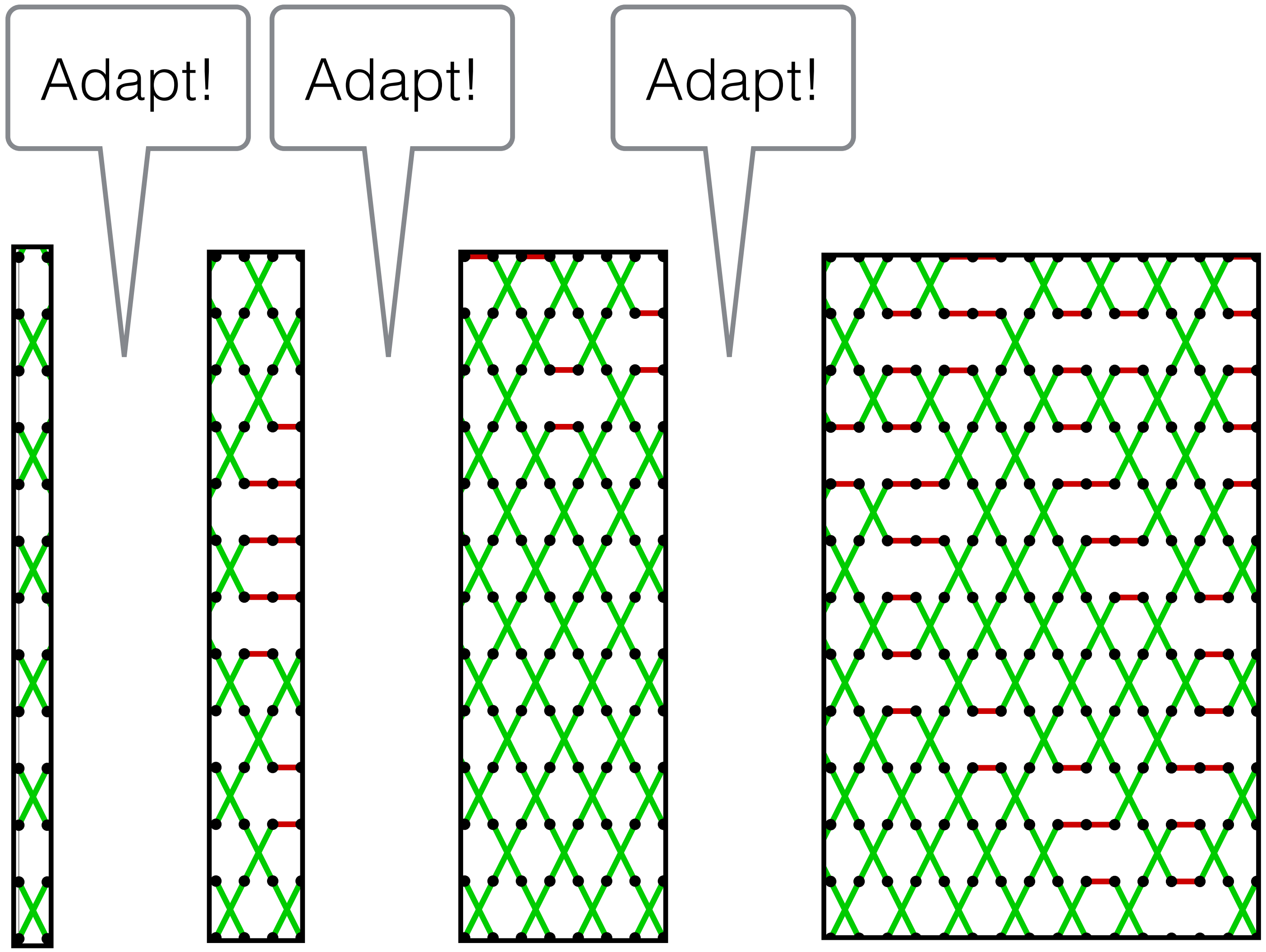


From performance models to algorithm tuning

Tuning nobs

- Annealing schedule
  - $0 = \beta_1 < \beta_2 < \ldots < \beta_N = 1$
  - number of grid points N
- “Pseudo-prior” $p(\beta)$ [for ST only]

Round-based tuning



...

round	1	2	3	4
swaps in round	2^1	2^2	2^3	2^4

Turning a schedule

- Recall our formula for the Tour Effectiveness

$$\mathrm{TE}_{\mathrm{NR}}(\{\beta_i\}) = [1 + 2\sum_i (r(\beta_i, \beta_{i+1})^{-1} - 1)^{-1}]^{-1}$$

- Goal: optimize $\mathrm{TE}_{\mathrm{NR}}$ over $\{\beta_i\}$ for next round
- Can we "predict" the rejection rates $r(\beta_i, \beta_{i+1})$ for a putative schedule?

Taylor expansions

of rejection rates

$$r(\beta, \beta') = \int_{\beta}^{\beta'} \lambda(t) \mathrm{d}t + O(|\beta' - \beta|^k)$$

"Local communication barrier"

- For PT: $\lambda_{\mathrm{PT}}(\beta) = \frac{1}{2}\mathbb{E}|\ell(X_\beta) - \ell(X'_\beta)|, k = 3$ Syed et al., 2021
- For ST*: $\lambda_{\mathrm{ST}}(\beta) = \frac{1}{2}\mathbb{E}|\ell(X_\beta) - \mathbb{E}[\ell(X_\beta)]|, k = 2$ Biron et al., 2023+

$X_\beta, X'_\beta \sim \pi_\beta$

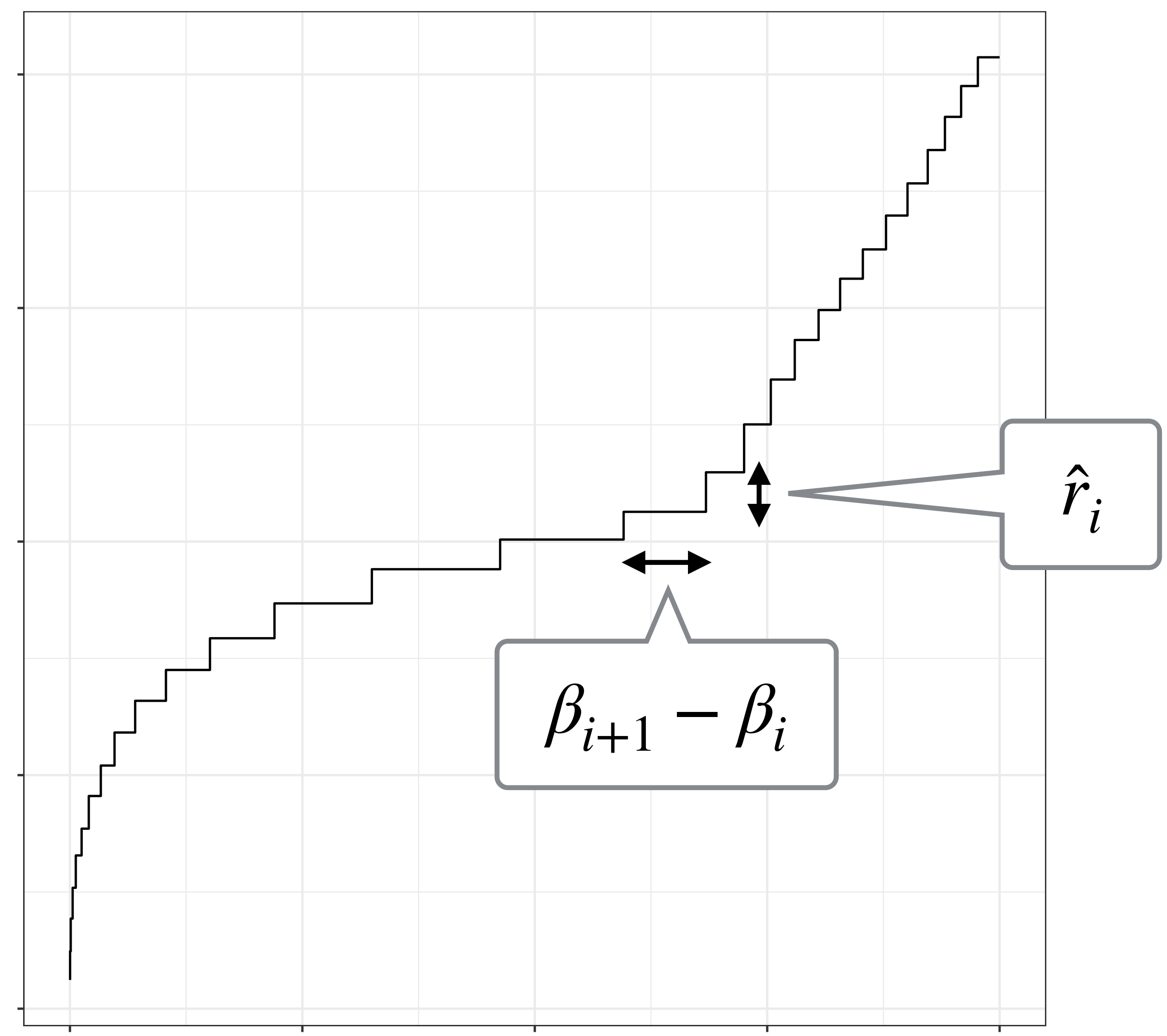
*when $p(\beta) \propto 1$

Estimating the communication barrier

Equivalently: estimate **cumulative** barrier $\boldsymbol{\Lambda}(\beta) = \int_0^\beta \lambda(t)\mathrm{d}t$

1. Start with initial schedule $\{\beta_i\}$, run PT for *n* iteration

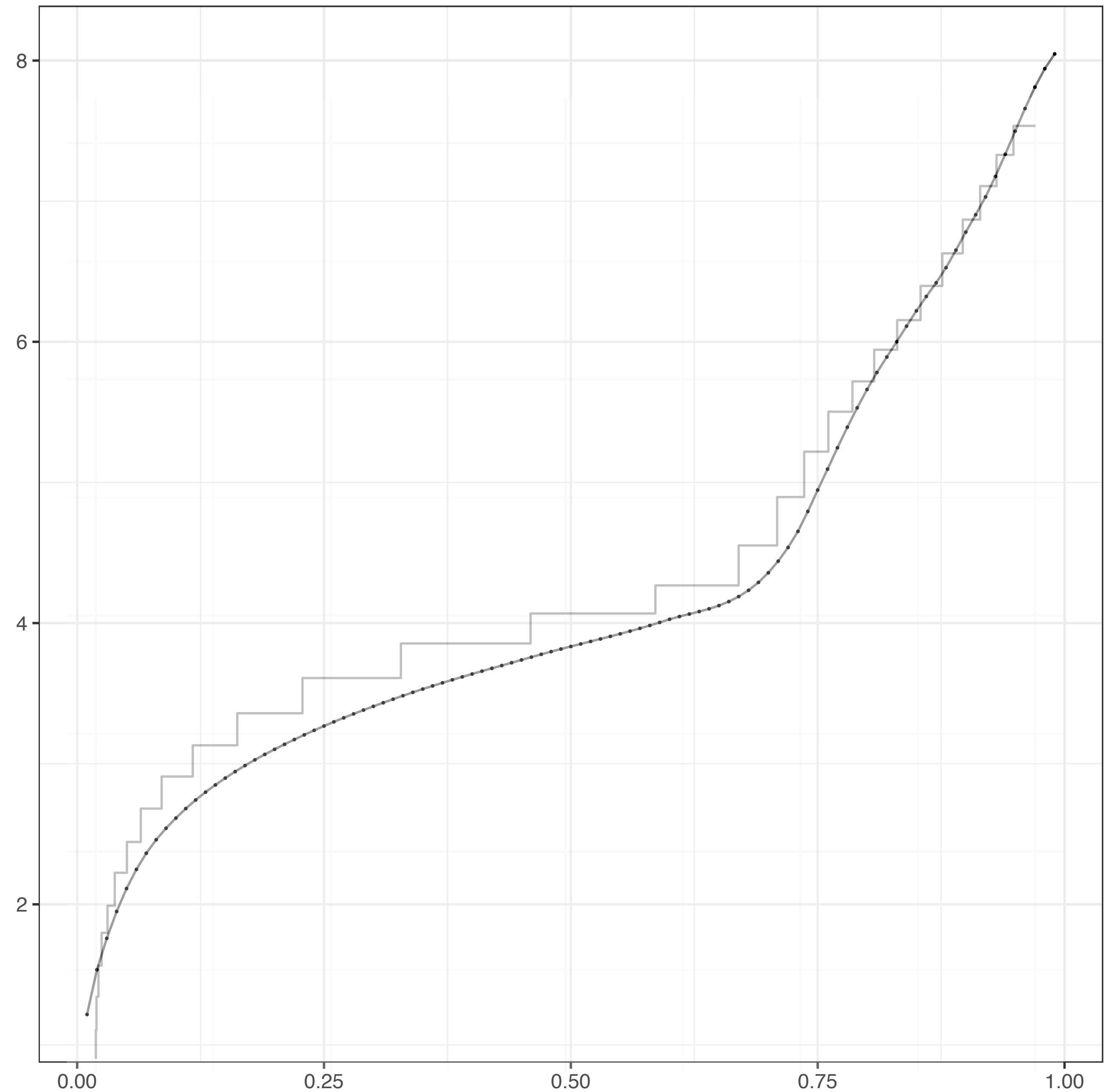
2. compute the following cumulative swap rejection statistics:



Syed et al., 2021

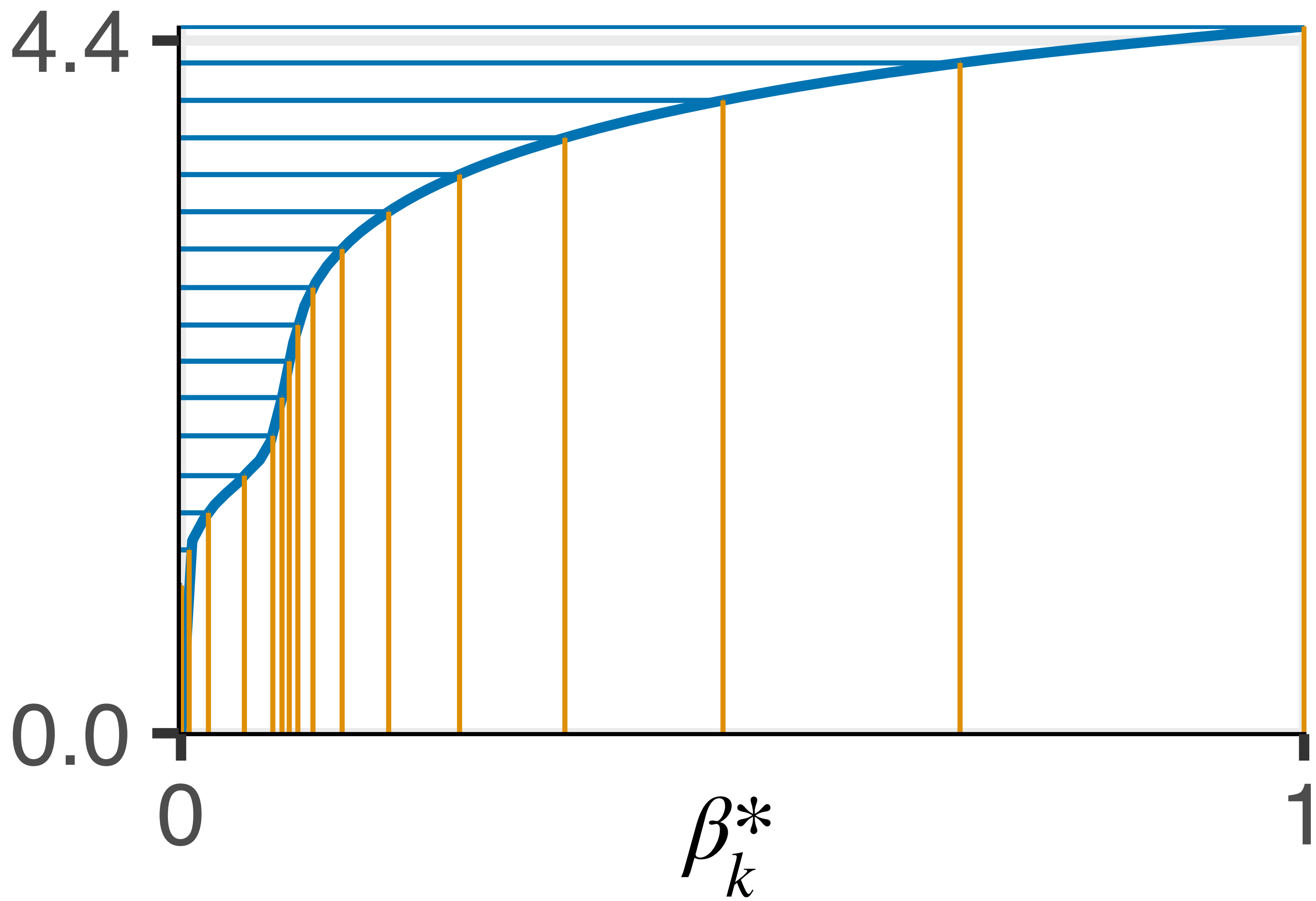
Estimating the communication barrier

3. Fit a monotone spline interpolation



Tuning the annealing schedule

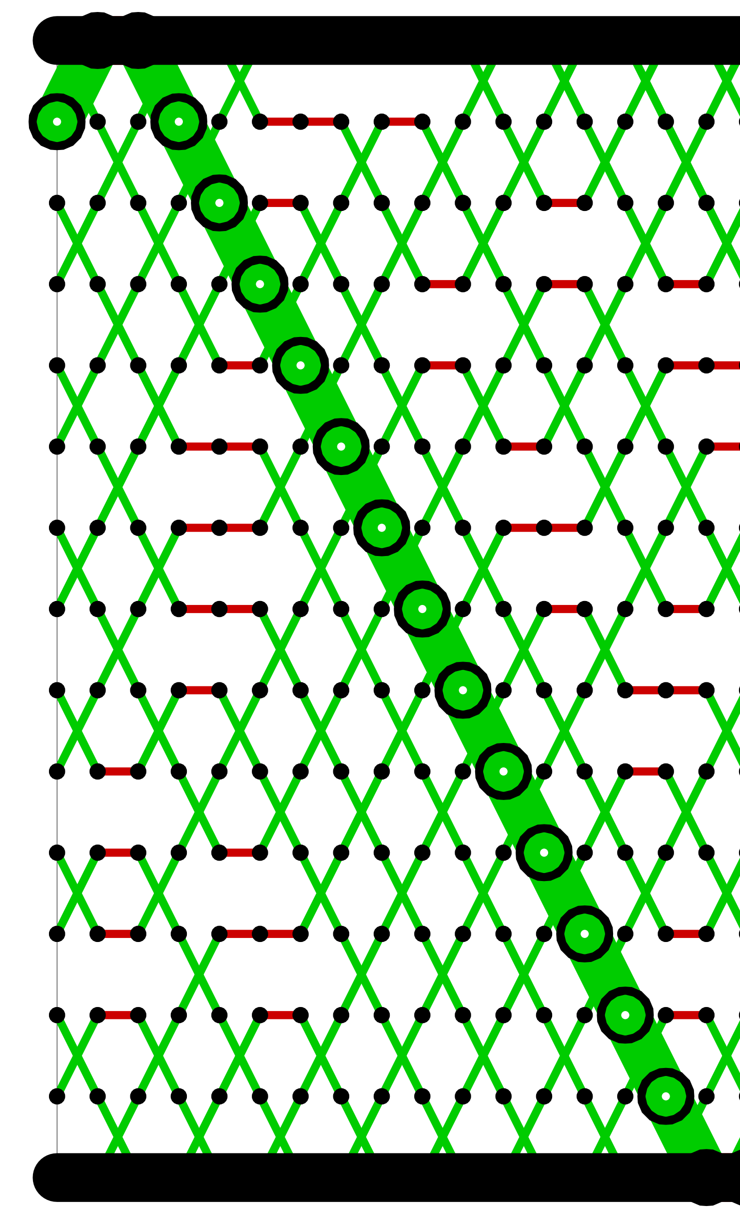
Close-form solution for $\max \text{TE}(\{\beta\})$ is given by



$$\beta_k^* = \Lambda^{-1}\left(\frac{k\,\Lambda(1)}{N}\right)$$

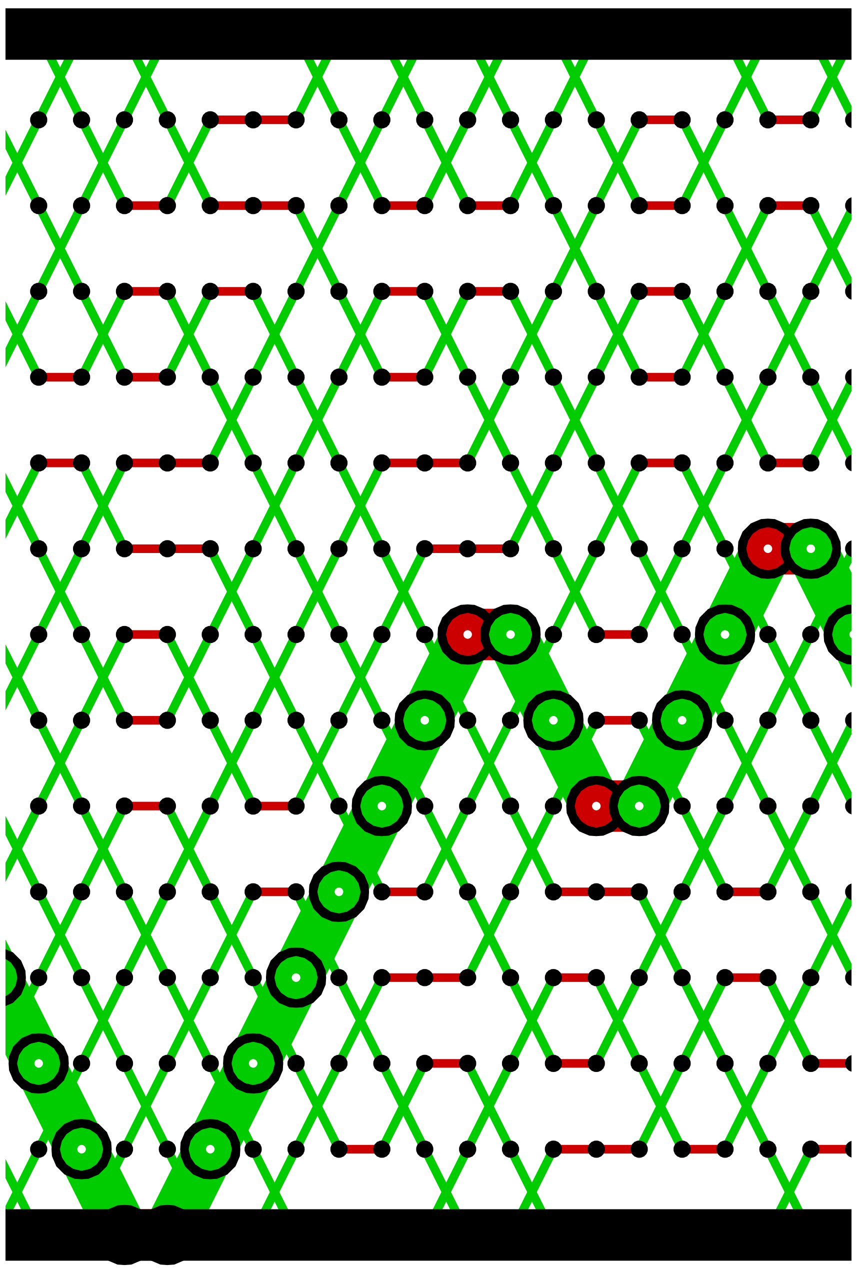
PDMP interpretation

- Let $N \to \infty$ (and $\beta_k = k/N$ for simplicity)
- From non-reversibility (of either ST or PT) emerges a persistence of motion or inertia....
- ....as long as a proposed swap is not rejected
- Swap rejection rate $r_i \to 0$...
  - ....but there are more and more chains to traverse → law of rare events → PDMP



Syed et al., 2021

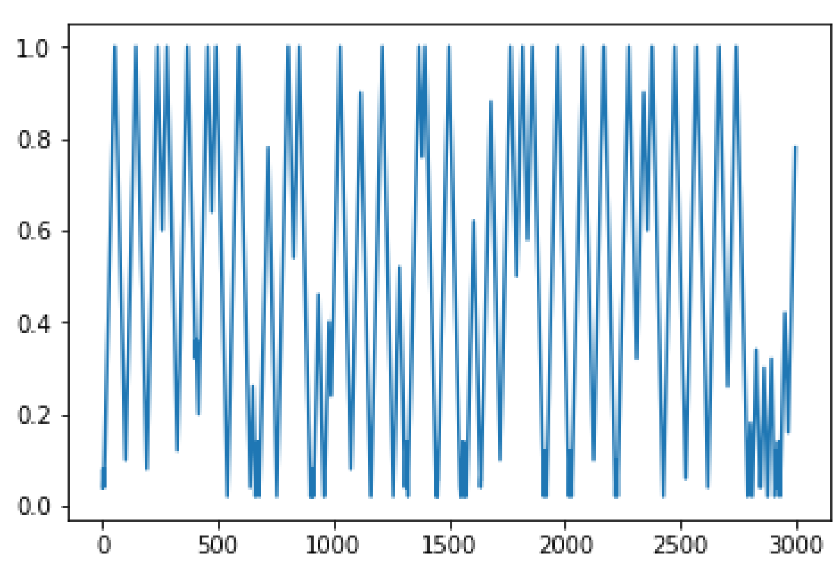
PDMP interpretation

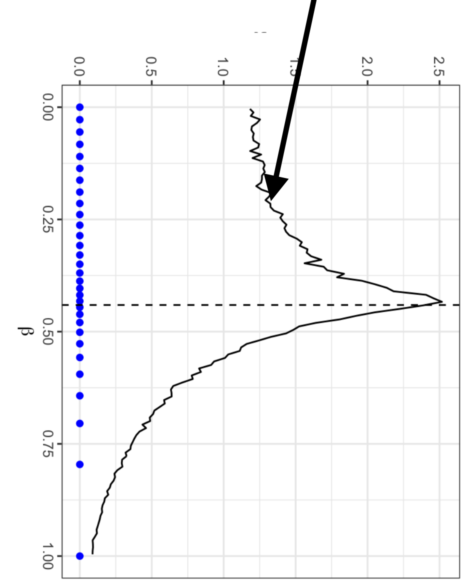


- As soon as there is a rejected swap, direction changes
- 1-dimensional “bounce”

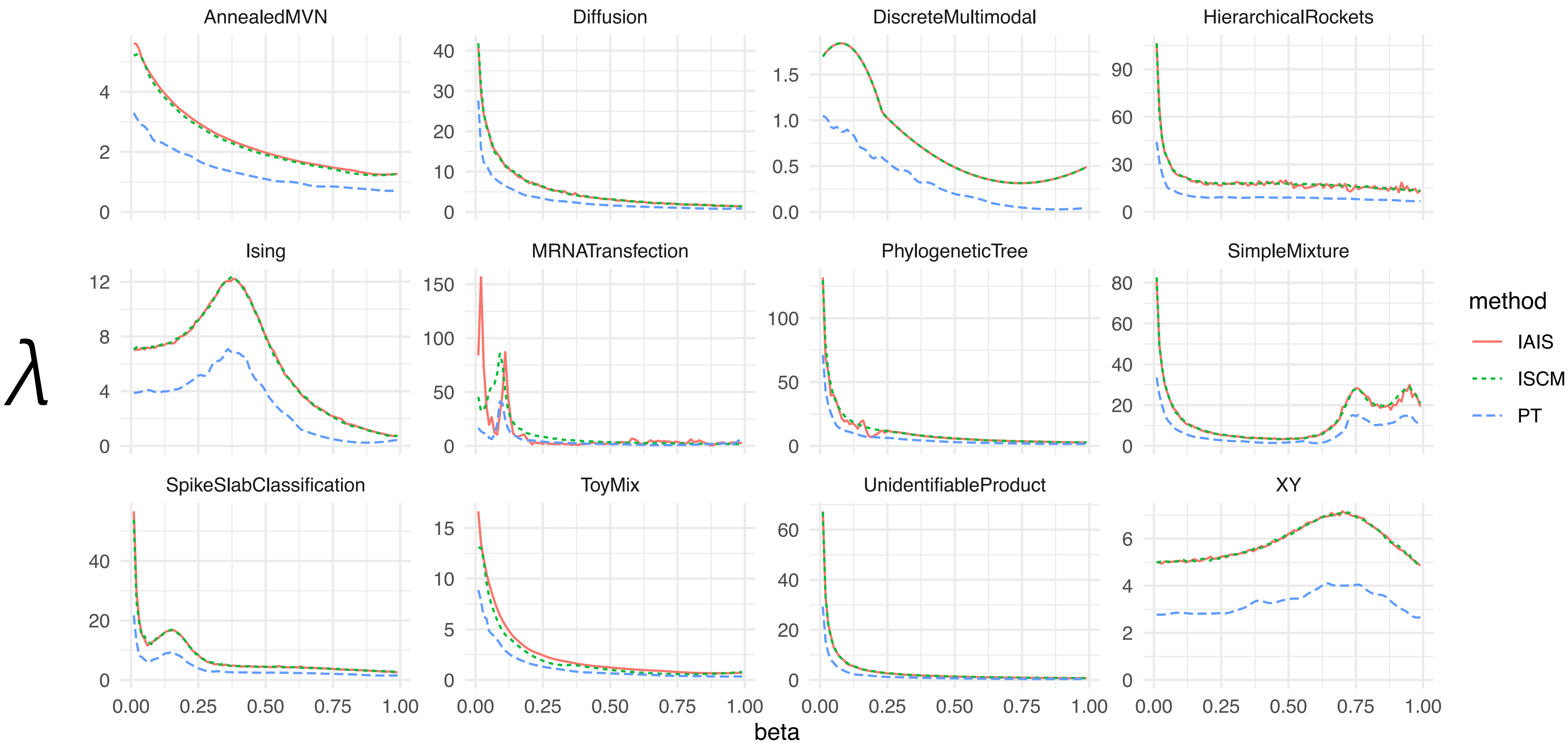
PDMP interpretation

- Under ELE, convergence to the "telegrapher PDMP"
  - velocity $\in \{-1, +1\}$
  - transition: velocity flip
  - rate: local communication barrier $\lambda(\beta)$





Local barrier: examples



Current work: generalizing the local barrier from PT to other “sequential change of measure” frameworks (ST, AIS, annealed SMC, flows, etc)

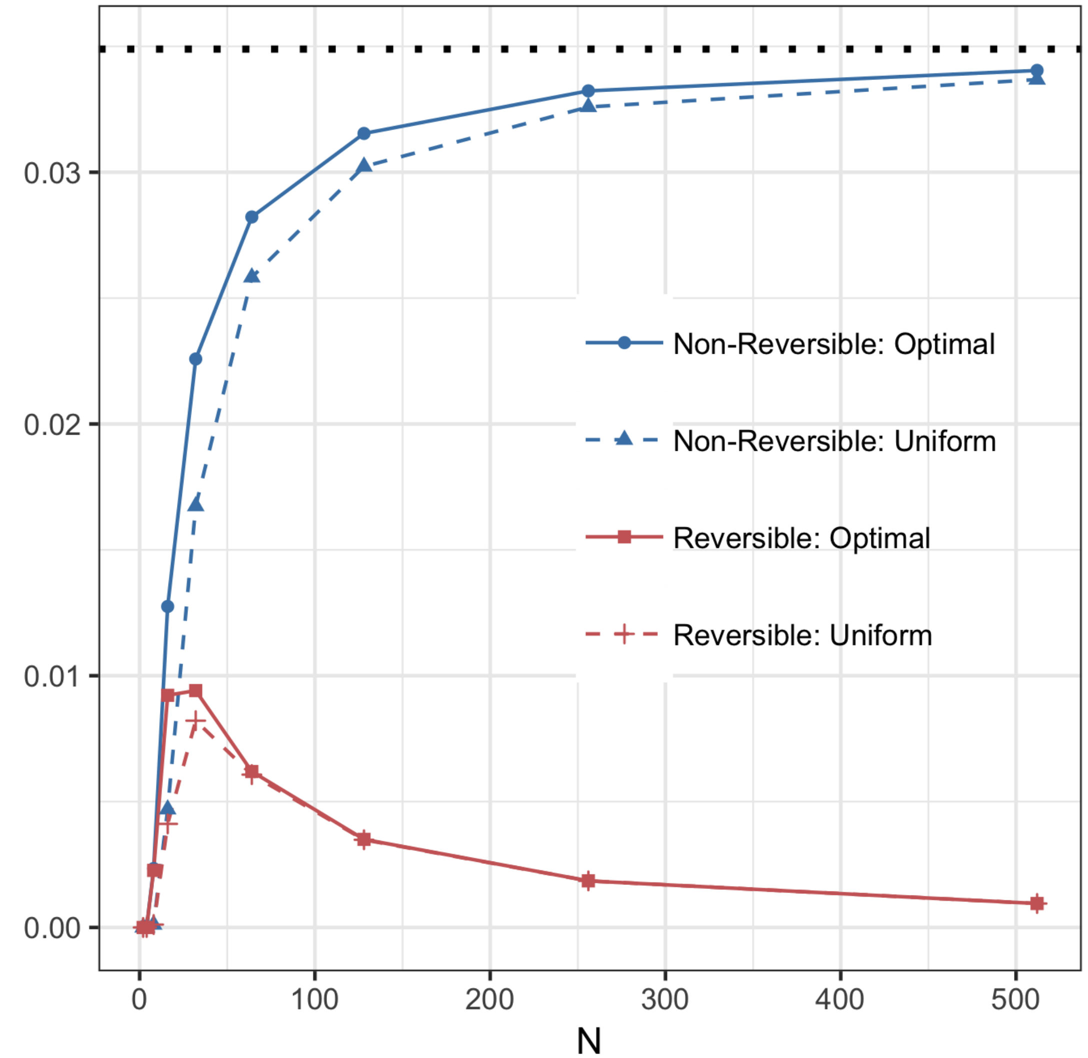
Communication barrier and TE

Under the ELE model

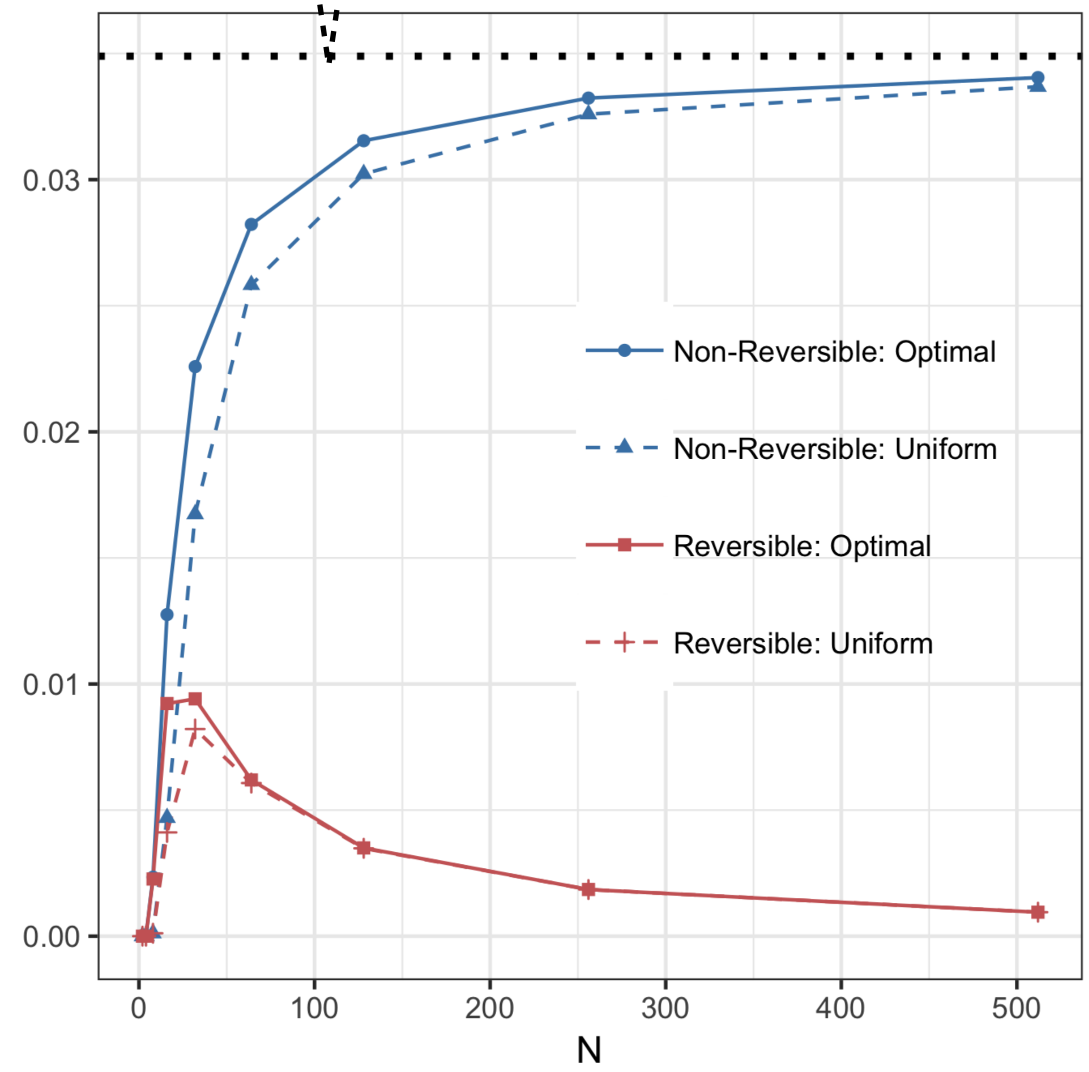
$$\mathrm{TE}_{\mathrm{NR}} \to \frac{1}{1 + 2\Lambda(1)}$$

$$\mathrm{TE}_{\mathrm{R}} \to 0$$

as $N \to \infty$



Can we “break” the barrier?



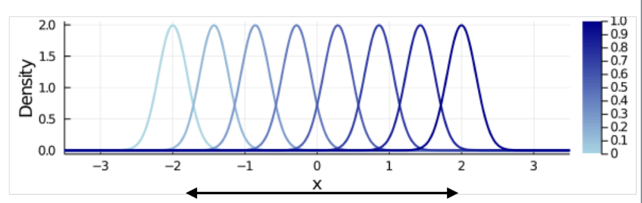
Breaking the communication barrier

- We have explored several strategies to break the barrier:
  - Smarter interpolation / geodesics (Syed et al, ICML 2021)

Standard annealing path:

$$\pi_t \propto \pi_0^{1-t}\pi^t$$

Proposition: $\bar{\tau} = \Theta(1/z)$

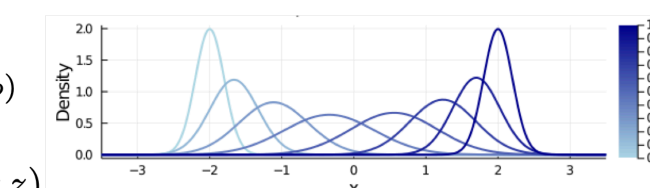


$z = |\mu_1 - \mu_0|/\sigma$

Generalized path:

e.g.: $\pi_t \propto \pi_0^{f_0(t,\phi)}\pi_1^{f_1(t,\phi)}$

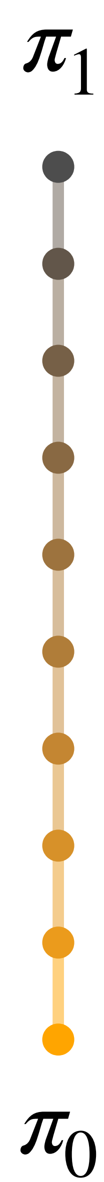
Proposition: $\bar{\tau} = \Omega(1/\log z)$

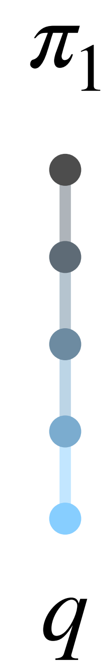


  - **Making the end point closer**
    (Surjanovic et al., NeurIPS 2022)

Variational-MCMC blends

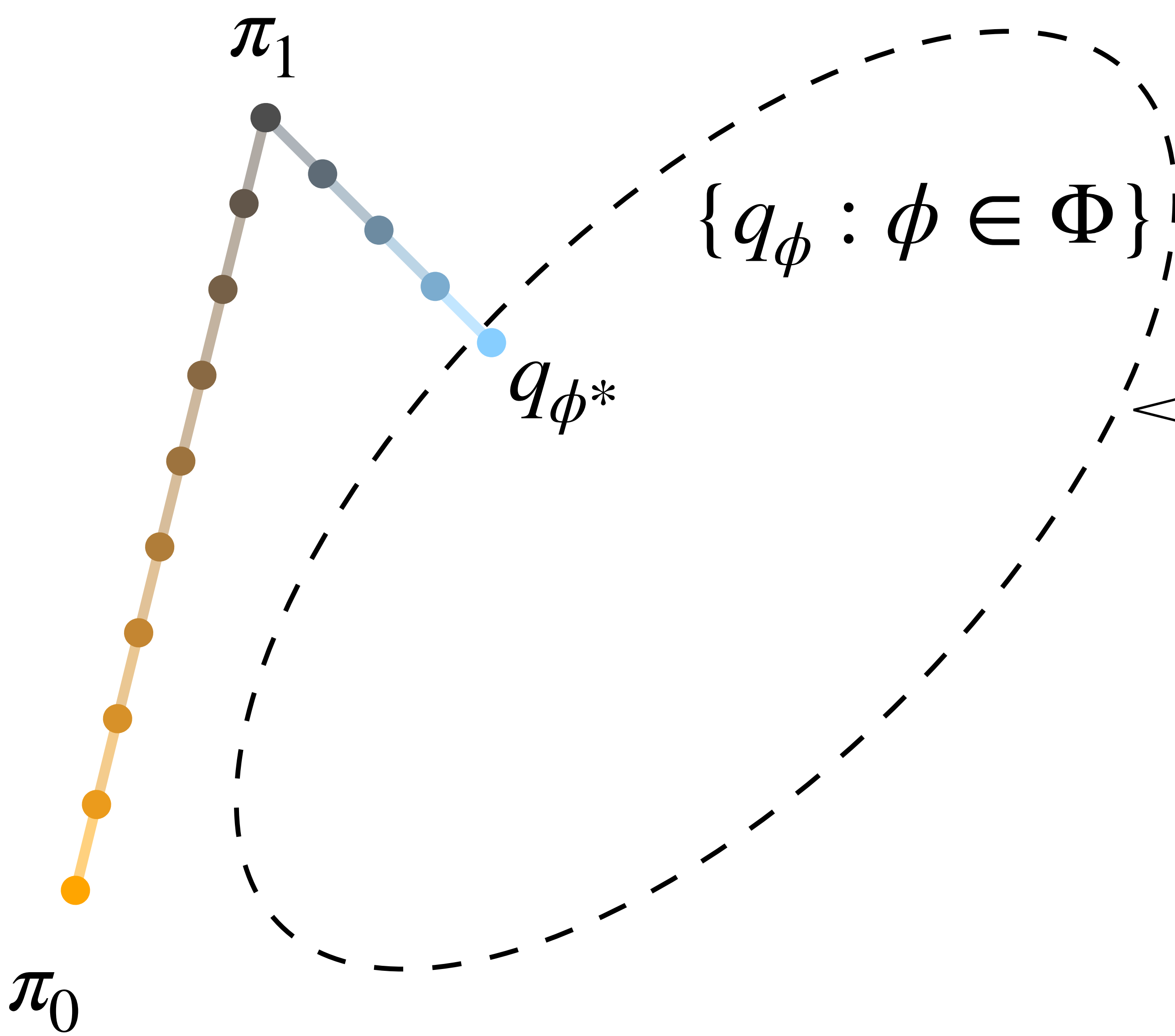
Bringing the end-point of the path closer to the target





How to find q close to π_1?

A family of easy distributions



Variational family

Example: Gaussian family

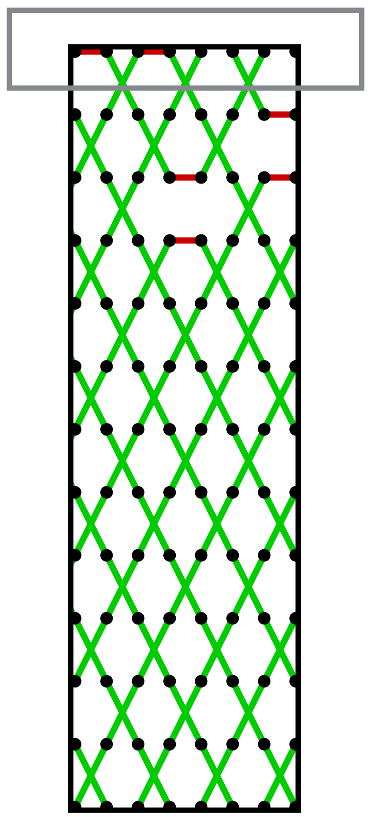
Generally: Parametric family such that we can:
- sample i.i.d.
- eval density

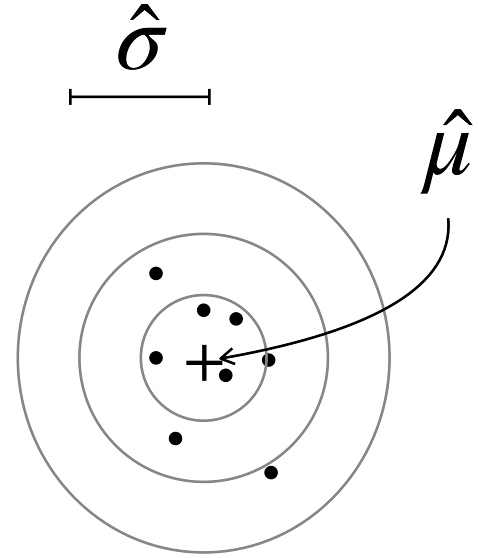
Paquet et al., 2009

Variational inference via statistical estimation

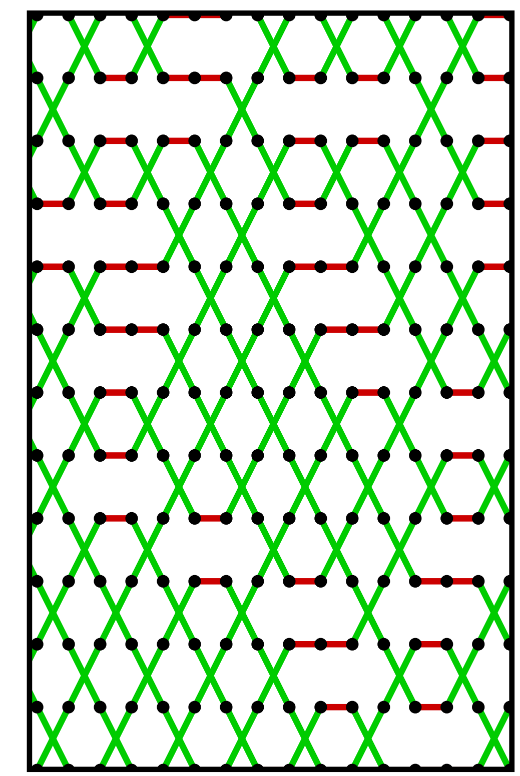
Reference samples
$X_1, X_2, \ldots, X_8$

...





Match
moments



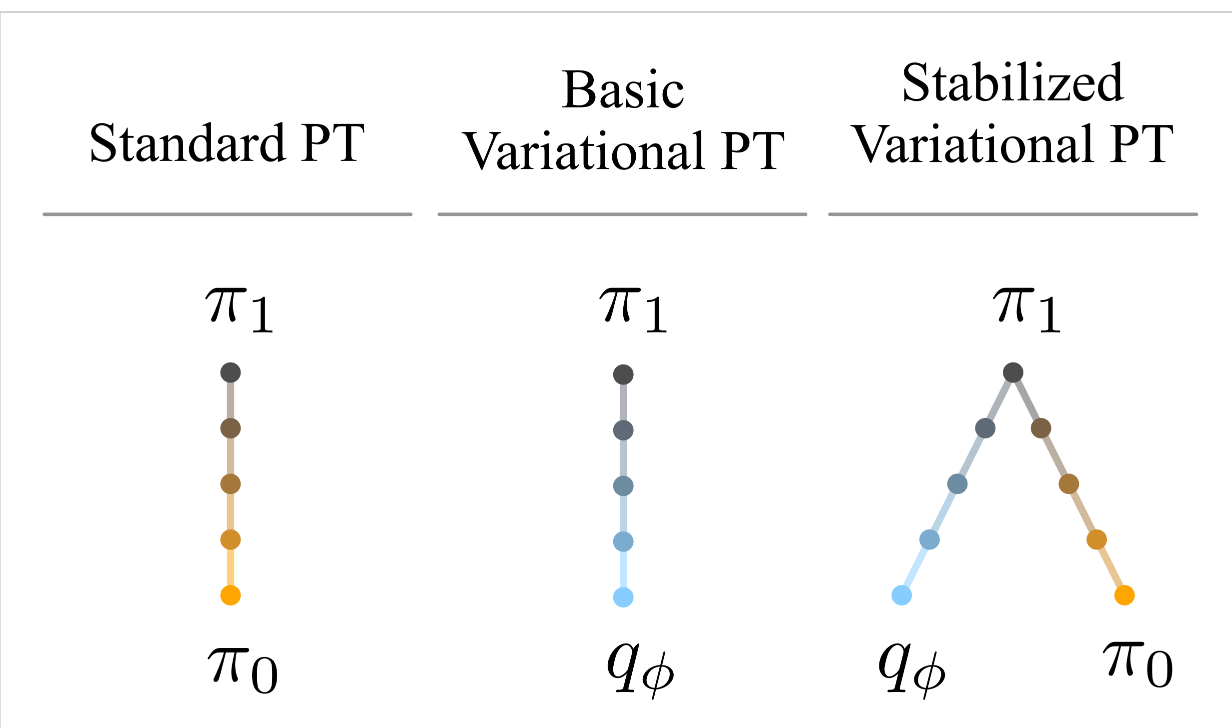
...

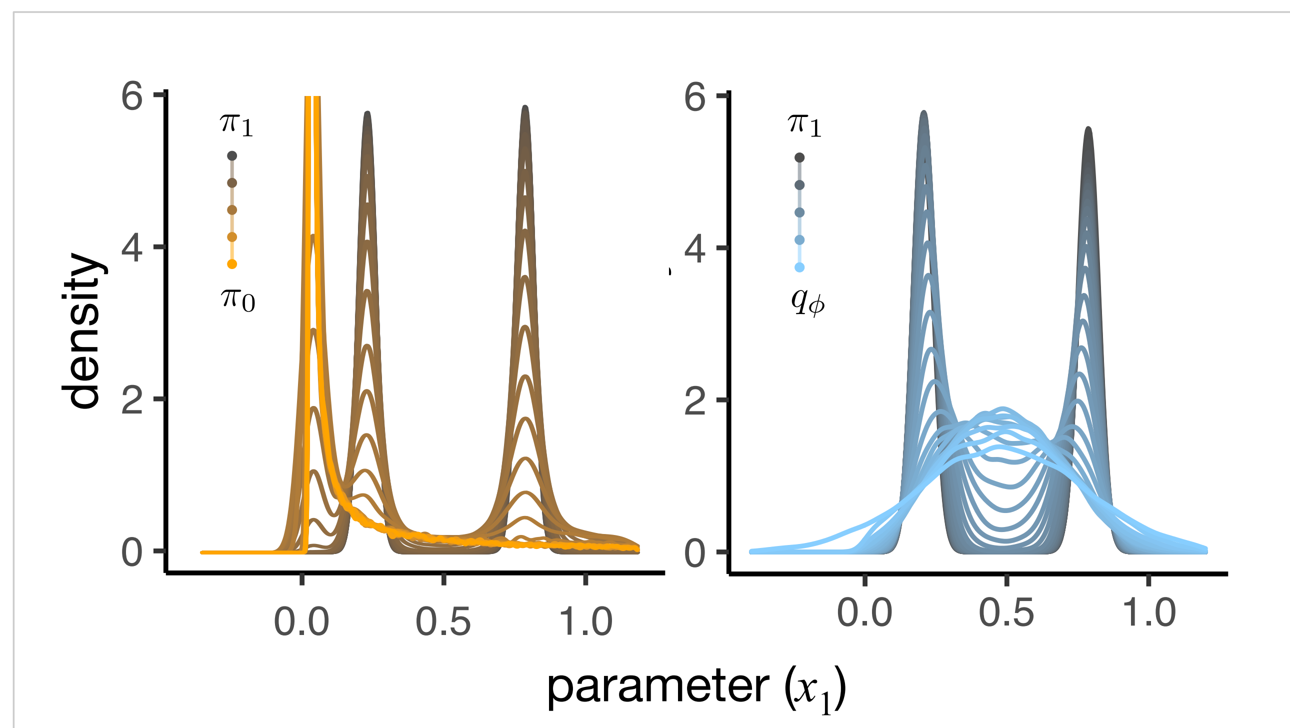
round	3	$q_\phi \leftarrow q_{(\hat{\mu},\hat{\sigma})}$	4
swaps in round	$2^3 = 8$	Update reference	2^4

Surjanovic et al, 2022

Stabilized Variational PT

- Performance of previous variational PT methods can collapse
- Developed a “stabilized” variational algorithm
  - non-deterioration guarantee under ELE






Open software implementation

Pigeons.jl

https://tinyurl.com/getpigeons

- In Quebec: “Pigeons.” = “Let us draw at random.”
- What Pigeons.jl does:
  - run Variational PT on your laptop
  - ... or on a cluster with 1000s of machines (MPI)

Nikola Surjanovic

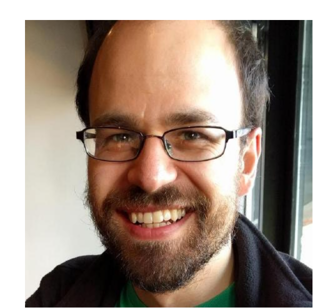
Miguel Biron

Paul Tiede

Saifuddin Syed

Trevor Campbell

Alexandre Bouchard-Côté



How to use Pigeons.jl

```
julia> using Pkg; Pkg.add("Pigeons"); using Pigeons # install
...
julia> approximation = pigeons(target =                    )
```

- Support many ways to specify the target distribution
  - Plain Julia function
  - Bayesian modelling languages (Turing.jl, Stan, etc)
  - "Black box" MCMC explorers in any language! (R, python, C, scala, ..)

Toy example

Communication barrier

Swap probability

```
julia> approximation = pigeons(target = toy_mvn_target(100), n_chains = 20)
────────────────────────────────────────────────────────────────────────────────────
  #scans      rd-trip    restarts       Λ        time(s)     log(Z)     min(α)     mean(α)
────────── ────────── ────────── ────────── ────────── ────────── ────────── ──────────
        2          0          0       9.77   8.38e-05       -118   4.99e-06      0.486
        4          0          0       10.4   0.000233       -114     0.0152      0.452
        8          0          0       7.21   0.000369       -113      0.307      0.621
       16          0          0       8.16    0.00193       -115      0.137       0.57
       32          0          0       8.58   0.000831       -115      0.162      0.549
       64          0          0       8.26    0.00127       -115      0.383      0.565
      128          0          0        8.8    0.00218       -115      0.403      0.537
      256          0          4       8.61     0.0043       -115      0.496      0.547
      512          7         11       8.66     0.0113       -115       0.47      0.544
 1.02e+03         19         31       8.62     0.0155       -115      0.507      0.547
────────────────────────────────────────────────────────────────────────────────────
PT(checkpoint = false, ...)
```

Round structure

Estimation of normalization constants

Distributed PT

1M dimensional target x 1k chain = 1B dim MCMC
1000 MPI processes

```
julia> job = pigeons(target = toy_mvn_target(1_000_000), n_chains = 1000,
                    on = MPI(n_mpi_processes = 1000))
```

Distributed PT

1M dimensional target x 1k chain = 1B dim MCMC
1000 MPI processes

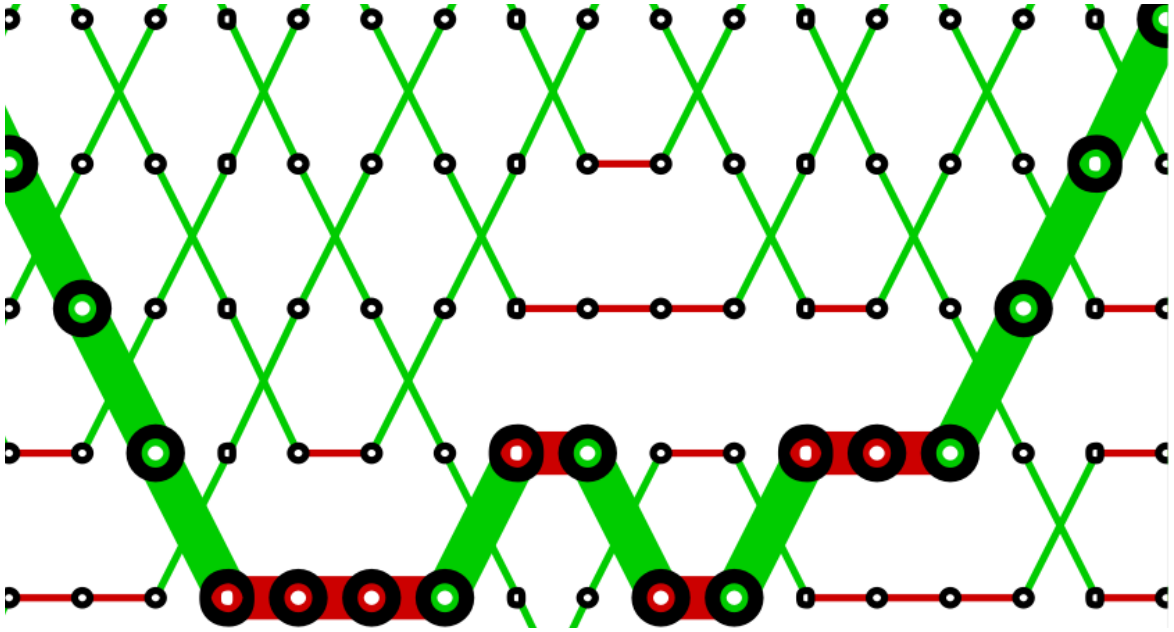
```
julia> job = pigeons(target = toy_mvn_target(1_000_000), n_chains = 1000,
                    on = MPI(n_mpi_processes = 1000))

Result{PT}(“/st-alexbou-1/abc/Pigeons/results/all/2023-05-19-20-05-36-RGkcWuaI")

julia> watch(job)
────────────────────────────────────────────────────────────────────────────
  #scans         Λ          time(s)     allc(B)     log(Z)      min(α)      mean(α)
────────── ────────── ────────── ────────── ────────── ────────── ──────────
        2        642        1.22    7.98e+03  -1.15e+06   8.89e-22       0.357
        4        642        1.21    1.39e+04  -1.15e+06   3.74e-25       0.357
        8        706       0.104    2.81e+04  -1.15e+06   3.26e-10       0.294
       16        724       0.208    4.92e+04  -1.15e+06   8.02e-06       0.276
       32        742       0.334    9.52e+04  -1.15e+06    0.00411       0.258
       64        745       0.553     1.8e+05  -1.15e+06     0.0554       0.254
      128        747        1.16     3.6e+05  -1.15e+06     0.0803       0.252
      256        750        2.25    6.98e+05  -1.15e+06      0.131        0.25
      512        749         4.6    1.38e+06  -1.15e+06      0.162        0.25
 1.02e+03        749        9.29    2.73e+06  -1.15e+06      0.189        0.25
────────────────────────────────────────────────────────────────────────────
```

The magic of distributed PT

- Exchange β's, not states!
  $\Longrightarrow$ visualize each MPI process as an **index process**



$\Longrightarrow$ O(1) network transmission/swap

- O(d log N) communication between rounds..
  .. but there are logarithmically many rounds

Thank you!

Links to papers

Syed et al. JRSSB 2021 https://tinyurl.com/nrpt-paper

Surjanovic et al. NeurIPS 2022 https://tinyurl.com/variational-paper

Link to software

BC et al., Julia Conf 2023 https://tinyurl.com/getpigeons