Mixed Hamiltonian Monte Carlo for Mixed Discrete and Continuous Variables
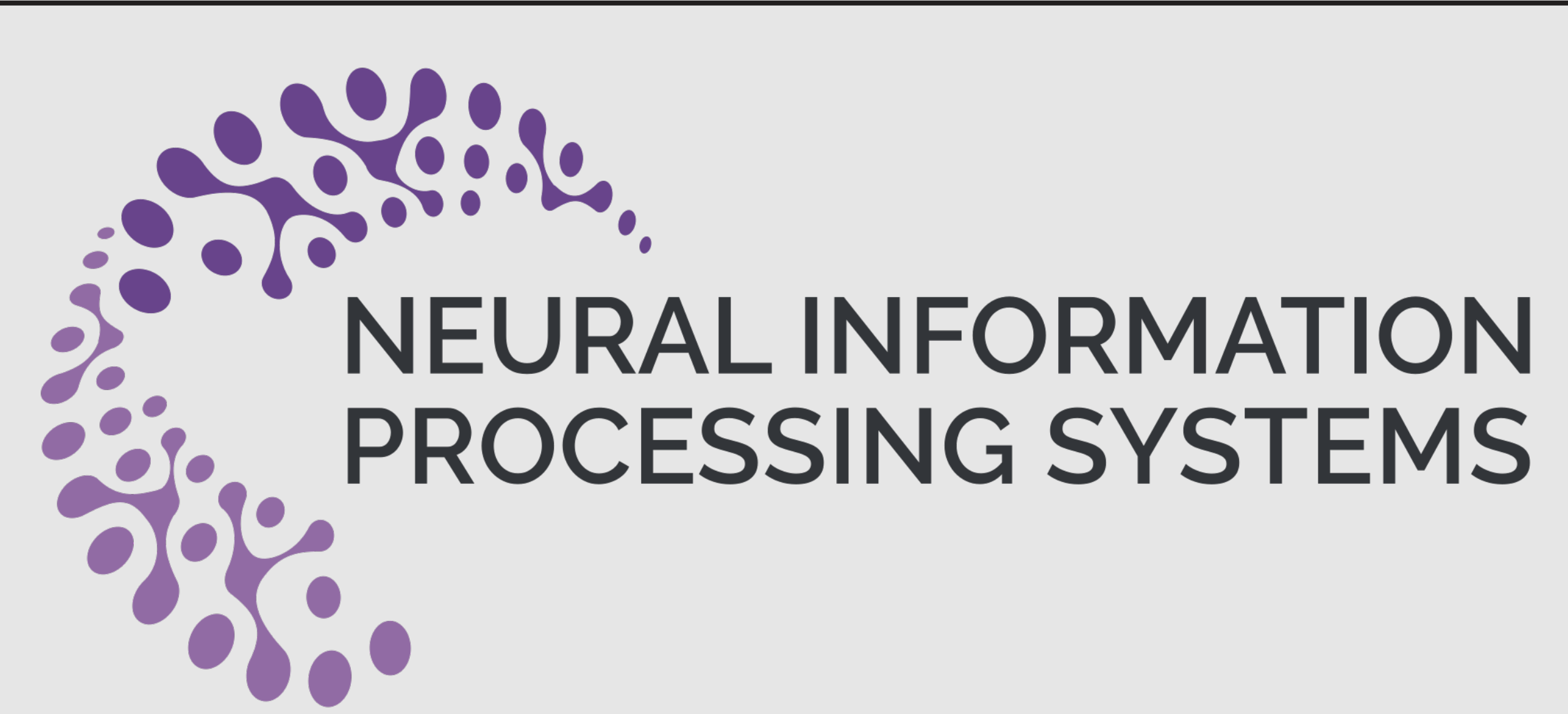
Guangyao Zhou @ *Vicarious AI*

stannis@vicarious.com

NEURAL INFORMATION PROCESSING SYSTEMS

1 Introduction

Hamiltonian Monte Carlo (HMC): sample from *continuous target distribution* $\pi(q^{\mathcal{C}}) \propto e^{-U(q^{\mathcal{C}})}$ by introducing auxiliary momentum variables $p^{\mathcal{C}} \in \mathbb{R}^{N_{\mathcal{C}}}$ and simulating *Hamiltonian dynamics*
$$\begin{cases} \frac{\mathrm{d}q^{\mathcal{C}}(t)}{\mathrm{d}t} = \nabla K^{\mathcal{C}}(p^{\mathcal{C}}) \\ \frac{\mathrm{d}p^{\mathcal{C}}(t)}{\mathrm{d}t} = -\nabla_{q^{\mathcal{C}}} U(q^{\mathcal{C}}) \end{cases}$$

Remarkable empirical success, but can't be applied to distributions with mixed discrete and continuous variables

Our Goal: sample from *target distribution* $\pi(x, q^{\mathcal{C}}) \propto e^{-U(x,q^{\mathcal{C}})}$ with mixed *discrete* ($x \in \Omega$) and *continuous* ($q^{\mathcal{C}} \in \mathbb{R}^{N_{\mathcal{C}}}$) variables

Existing approaches:

- Integrate out the discrete variables:
  - Only applicable on a small scale
  - Can't always be carried out automatically

    Stan — *Pyro/NumPyro*

- Alternate between continuous HMC and generic discrete updates:
  - Discrete updates can only be done infrequently
  - Hard to make long-range proposals

    PyMC3 — *Turing.jl*

- **Update discrete and continuous variables in tandem**:
  - *Discontinuous HMC (DHMC), Probabilistic path HMC (PPHMC)*
  - *DHMC is best suited for ordinal paramters, and has inefficient embedding and algorithmic structure*
  - *PPHMC only works for phylogenetic trees*

2 Mixed Hamiltonian Monte Carlo (M-HMC)

Mixed Hamiltonian Monte Carlo (M-HMC)

- M-HMC also evolves the discrete and continuous variables in tandem
- M-HMC is applicable to any distributions with mixed support
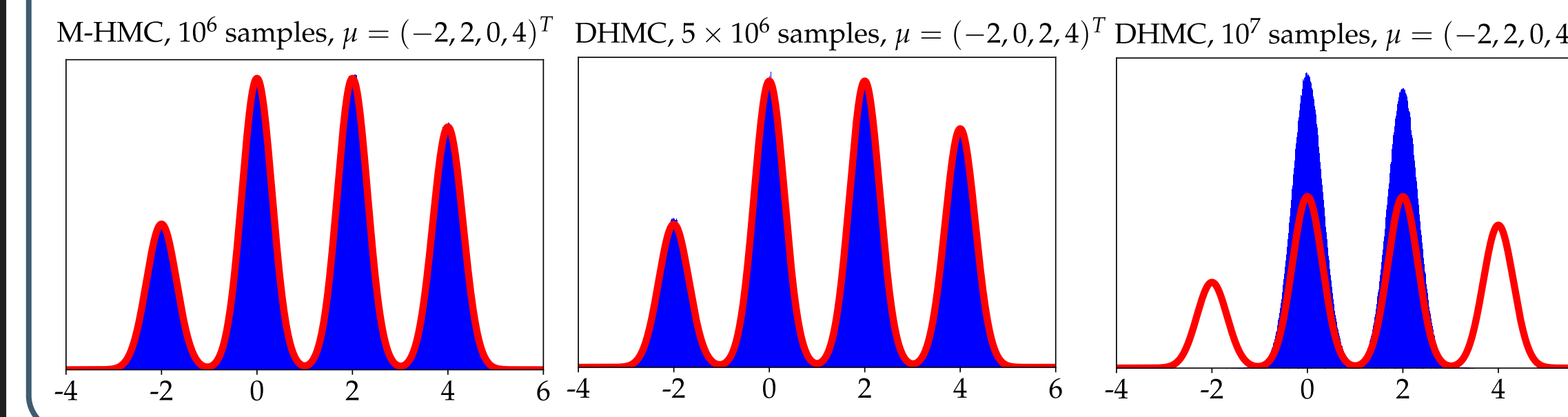- M-HMC can be efficiently implemented using Laplace momentum

We start with an illustrative example on a 1D Gaussian mixture model (GMM) with 4 mixture componets. Use $x \in \{1, 2, 3, 4\}$ to denote the discrete variable, and $q^{\mathcal{C}} \in \mathbb{R}$ to denote the continuous variable. We want to sample from
$$\pi(x, q^{\mathcal{C}}) = \phi_x N(q^{\mathcal{C}}|\mu_x, \Sigma)$$
where
$$\phi_1 = 0.15, \phi_2 = \phi_3 = 0.3, \phi_4 = 0.25, \Sigma = 0.1$$
$$\mu_1 = -2, \mu_2 = 0, \mu_3 = 2, \mu = 4$$

The right panel shows M-HMC in this simple case. The simple change implied by the M-HMC framework leads to a more efficient sampler that is able to correct the bias from naively doing Metropolis-Hastings (MH) updates within HMC.

More generally, M-HMC can be easily applied to arbitrary distributions with mixed support, and introduces minimal overhead compared to existing HMC methods. Refer to the paper for more details on the general algorithm.

Comparison with Discontinuous HMC (DHMC):

- DHMC relies on embedding that works best for ordinal parameters
  $$x_i = n \iff q_i^{\mathcal{D}} \in (a_n, a_{n+1}), 0 = a_1 \leq a_2 \leq \cdots$$
- DHMC needs to update all discrete variables at every step, inefficient

M-HMC, 10^6 samples, $\mu = (-2, 2, 0, 4)^T$ — DHMC, 5×10^6 samples, $\mu = (-2, 0, 2, 4)^T$ — DHMC, 10^7 samples, $\mu = (-2, 2, 0, 4)^T$

MHwHMC v.s. M-HMC for 1D GMM

```
Require: U, target potential
         Q, discrete proposal
Input:   x^(0), current discrete state
         q^(C(0)), current continuous location
         ε, step size; L, # of steps
function M-HMC(x^(0), q^(C(0)), ε, L|U, Q)
    k^(D(0)) ~ Exponential(1), p^(C(0)) ~ N(0,1)
    x ← x^(0), k^D ← k^(D(0))
    q^C ← q^(C(0)), p^C ← p^(C(0)), ΔU^D ← 0
    for t from 1 to L do
        q^C, p^C ← leapfrog(q^C, p^C, ε)
        x̃ ~ Q(·|x), ΔE ← log (e^{-U(x,q^C)} Q(x̃|x)) / (e^{-U(x̃,q^C)} Q(x|x̃))
```

\# Naive MH within HMC
```
        if Exponential(1) > ΔE then
            x ← x̃
        end if
```
Biased!!

or

\# M-HMC
```
        if k^D > ΔE then
            ΔU^D ← ΔU^D + U(x̃, q^C) − U(x, q^C)
            x ← x̃, k^D ← k^D − ΔE
        end if
    end for
```
Unbiased

\# Final MH correction
```
    E ← U(x, q^C) + K^C(p^C)
    E^(0) ← U(x^(0), q^(C(0))) + K^C(p^(C(0)))
    Accept x, q^C w.p. min{1, e^{-(E − E^(0) − ΔU^D)}}
    return x, q^C
end function
```

(a)

(b) MHwHMC, 10^7 samples, $\mu = (-2, 0, 2, 4)^T$

(c) M-HMC, 10^6 samples, $\mu = (-2, 0, 2, 4)^T$
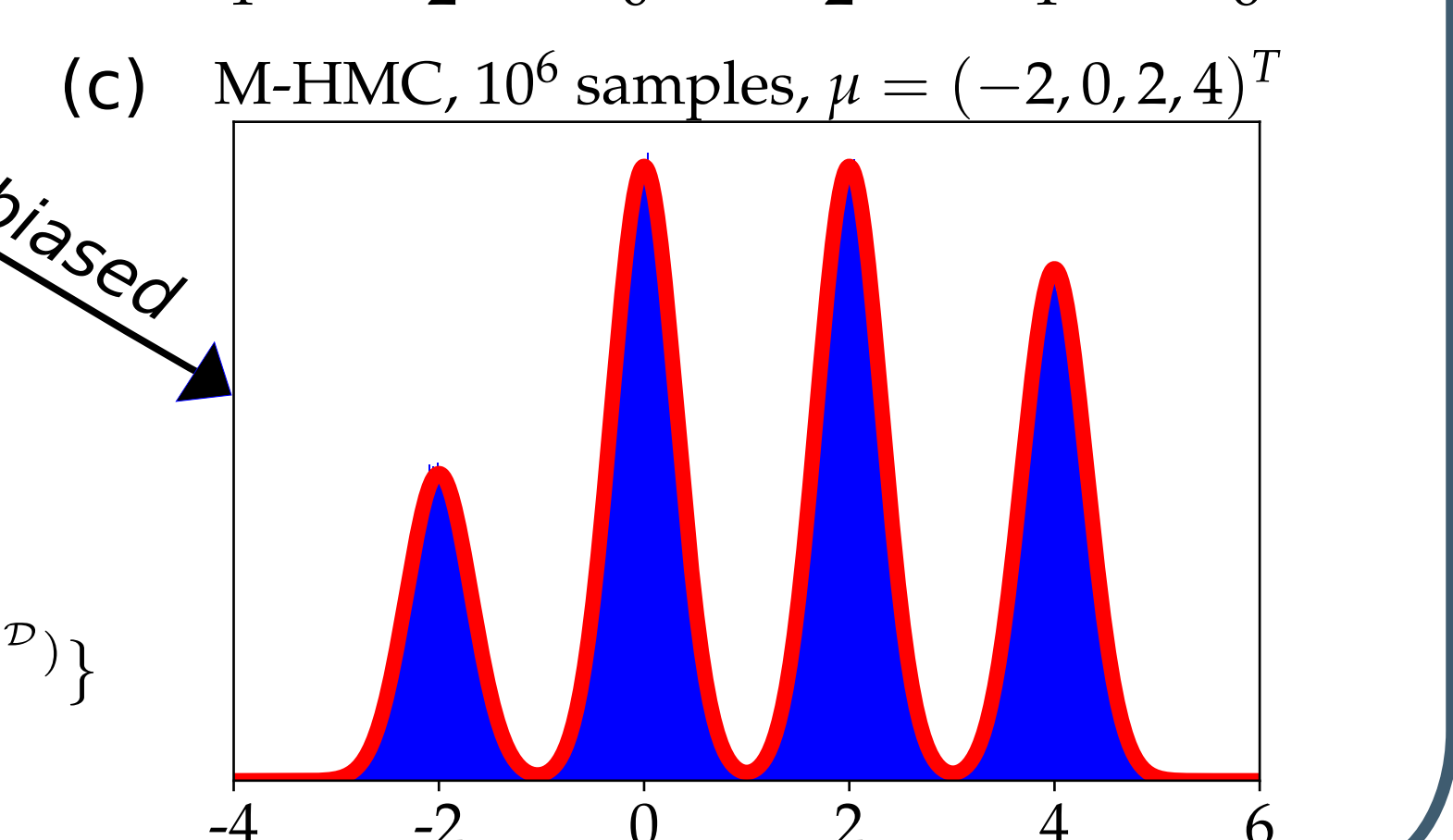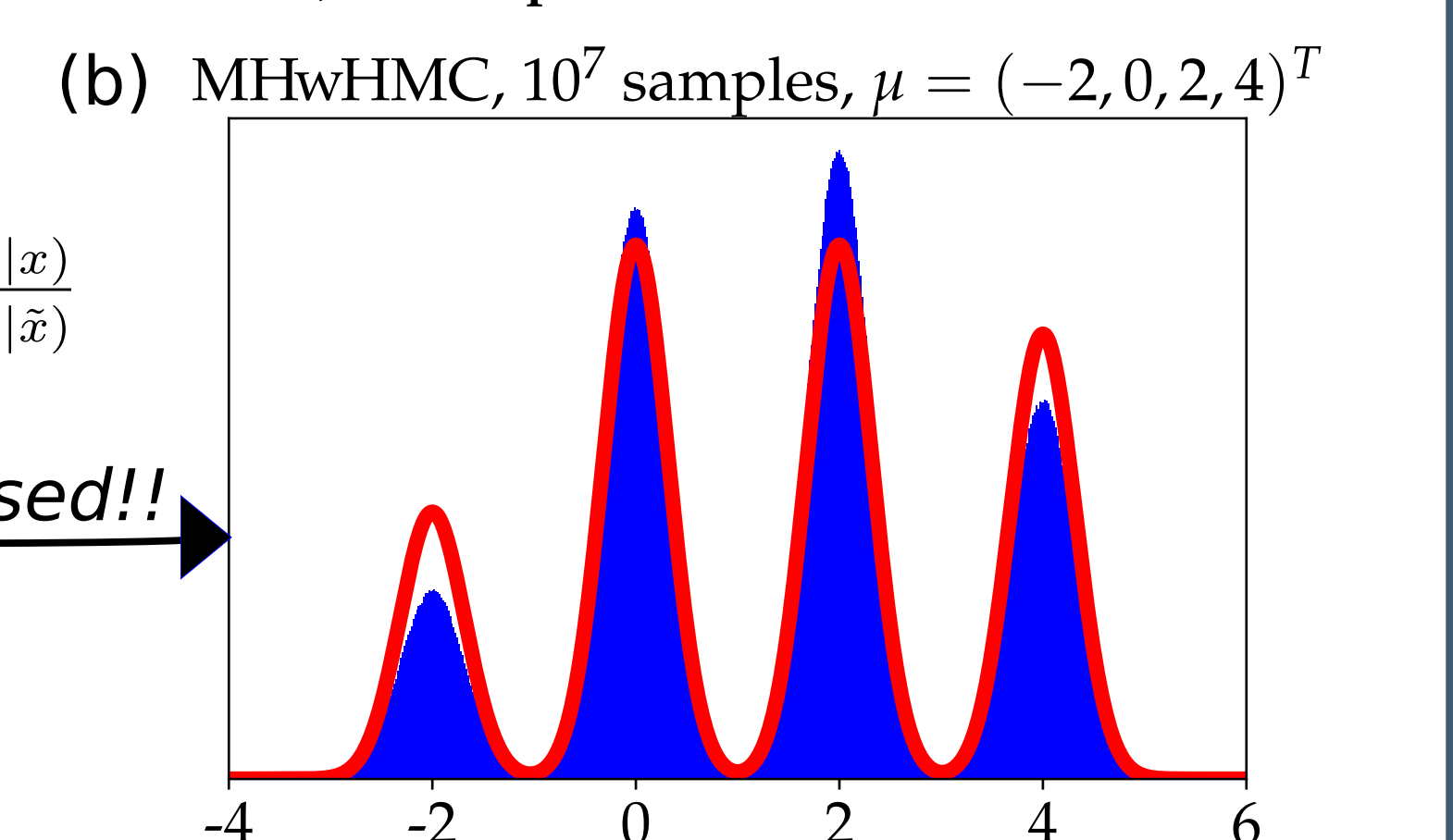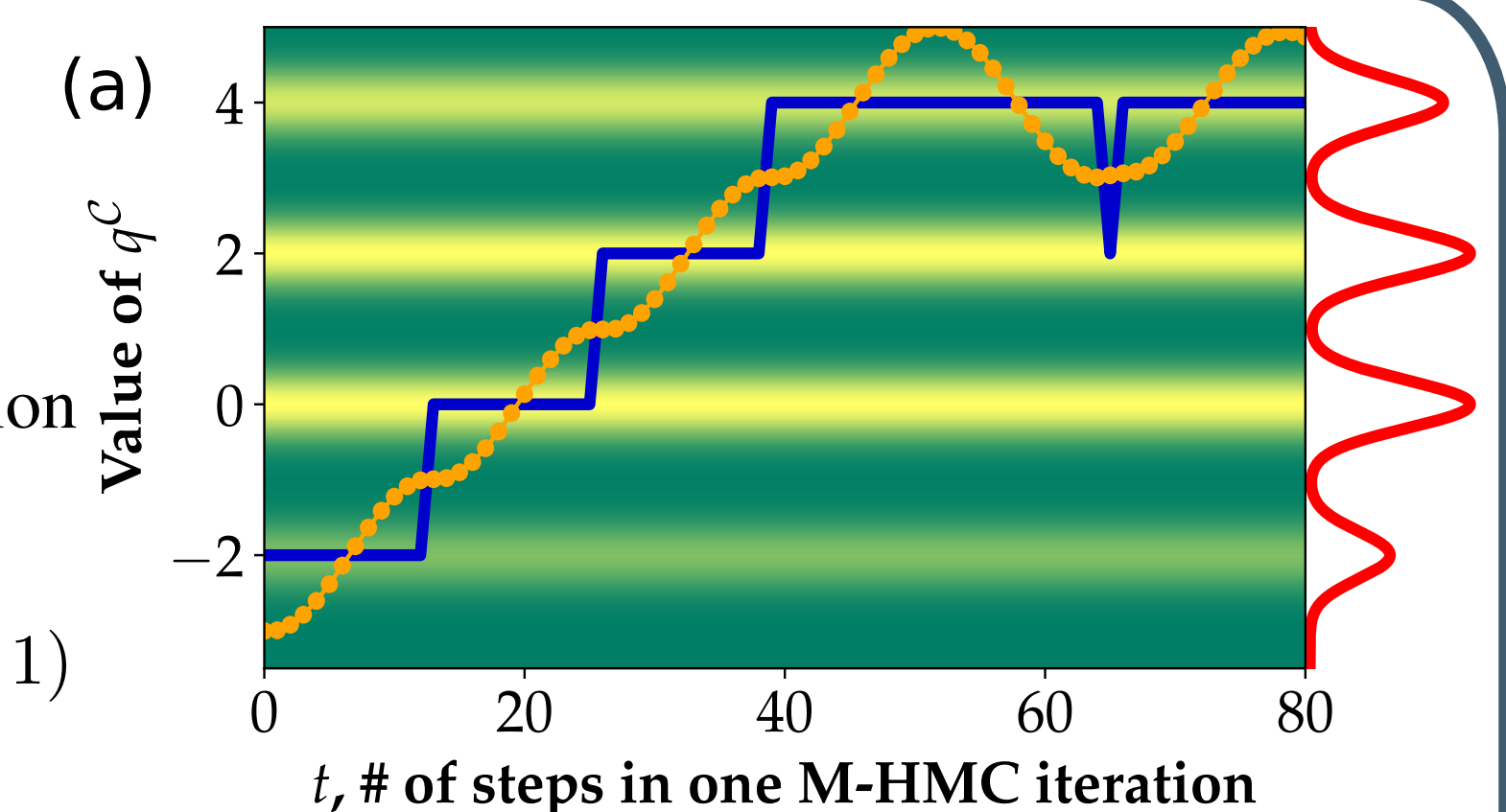
3 Numerical Experiments

Overview

Methods used:

Mixed HMC (M-HMC) : Custom *JAX* implementation

Discontinuous HMC (DHMC) : Custom *JAX* implementation

HMC-within-Gibbs (HwG) : Custom *JAX* implementation

No-U-Turn Sampler (NUTS) : Using *NumPyro*, for GMMs

NUTS-within-Gibbs (NwG) : Using compound step in PyMC3

Specialized Gibbs samplers : Custom *Numba* implementation

- (Polson et al., 2013) for Bayesian logistic regression
- (Chen et al., 2013) for Correlated topic models

Performance measure:

- Minimum relative effective sample size (MRESS)
- The minimum ESS over all dimensions
- Normalized by the number of samples
- Estimated using multiple independent chains

Discrete proposals in M-HMC:

$$Q_j(\tilde{x}|x) \propto \begin{cases} \pi(\tilde{x}, q^{\mathcal{C}}) & \text{if } \tilde{x}_i = x_i, i \neq j \\ 0 & \text{otherwise} \end{cases}$$

24D Gaussian mixture model (GMM)

Basic setup: $\pi(x, q^{\mathcal{C}}) = \phi_x N(q^{\mathcal{C}}|\mu_x, \Sigma)$ where
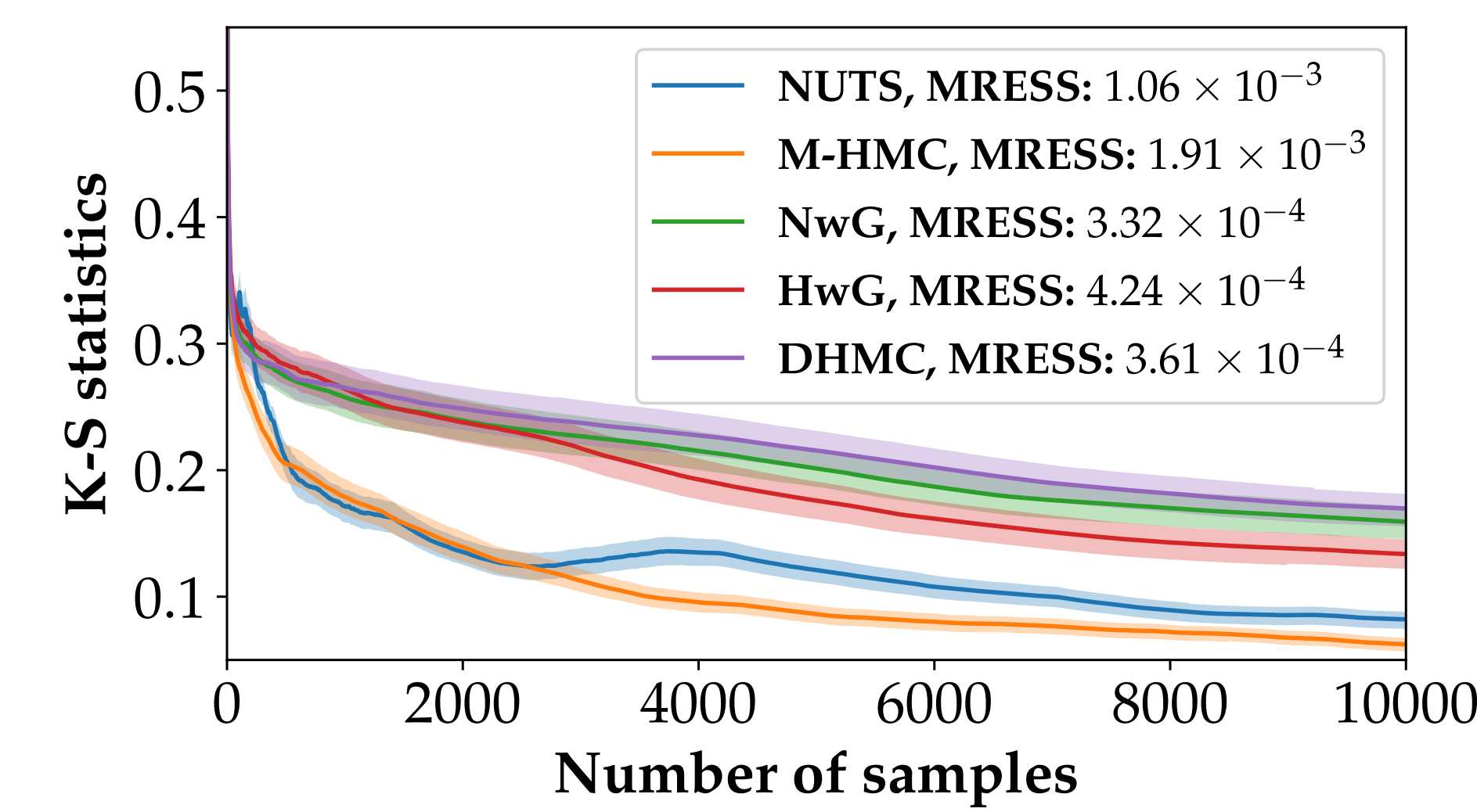$$\phi_1 = 0.15, \phi_2 = \phi_3 = 0.3, \phi_4 = 0.25, \Sigma = 3I$$
$$\mu_1, \mu_2, \mu_3, \mu_4 \text{ are from 24 permutations of } -2, 0, 2, 4$$

Target distribution: $\pi(x, q^{\mathcal{C}})$

Methods tested: M-HMC, DHMC, HwG, NwG, NUTS

Results summary:

- All samplers are accurate
- M-HMC is more efficient than DHMC, HwG and NwG
- M-HMC also outperforms NUTS

- NUTS, MRESS: 1.06×10^{-3}
- **M-HMC, MRESS: 1.91×10^{-3}**
- NwG, MRESS: 3.32×10^{-4}
- HwG, MRESS: 4.24×10^{-4}
- DHMC, MRESS: 3.61×10^{-4}

Variable selection in Bayesian logistic regression (BLR)

Basic setup: $y_i \sim Bernoulli\left(\sigma(X_i^T \beta)\right), i = 1, \cdots, 100$
where $X \in \mathbb{R}^{100 \times 20}, \beta \in \mathbb{R}^{20}$, and $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid function. $\gamma_j, j = 1, \cdots, 20$ are binary variables indicating the presence of a particular component of β, and $N(0, 25I)$ is an uninformative prior on β.

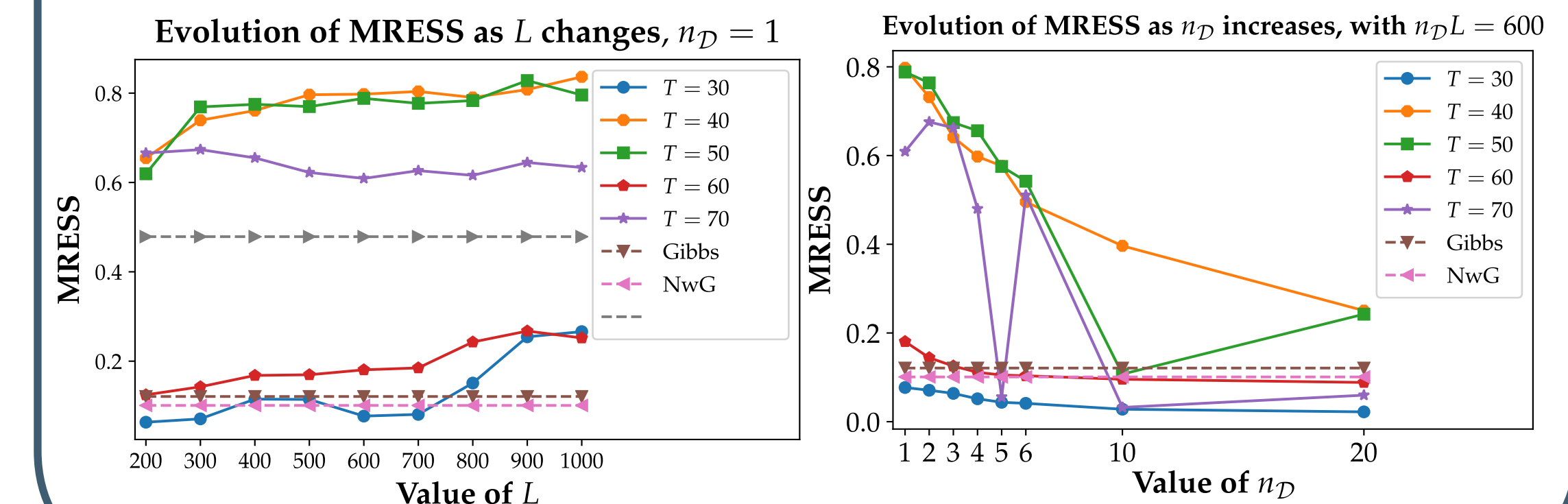Joint distribution: $p(\beta, \gamma, X, y) = N(\beta|0, 25I) \prod_{i=1}^{100} p_i^{y_i}(1 - p_i)^{1-y_i}$

where $p_i = \sigma(\sum_{j=1}^{20} X_{ij}\beta_j\gamma_j), i = 1, \cdots, 100$.

Target distribution: $p(\beta, \gamma|X, y)$

Methods tested: M-HMC, DHMC, HwG, NwG, Gibbs (Polson et al., 2013)

Results summary:

- All samplers are accurate
- M-HMC is more efficient than DHMC, HwG, NwG and Gibbs
- M-HMC exhibits U-turn behavior
- M-HMC benefits from distributed/more frequent discrete updates

Evolution of MRESS as L changes, $n_{\mathcal{D}} = 1$ — Evolution of MRESS as $n_{\mathcal{D}}$ increases, $n_{\mathcal{D}}L = 600$

Correlated topic models (CTMs)

Basic setup:
Given the topics β, a vector $\mu \in \mathbb{R}^K$ and a $K \times K$ covariance matrix Σ, CTM assumes the following generative process for the dth document with N_d words:
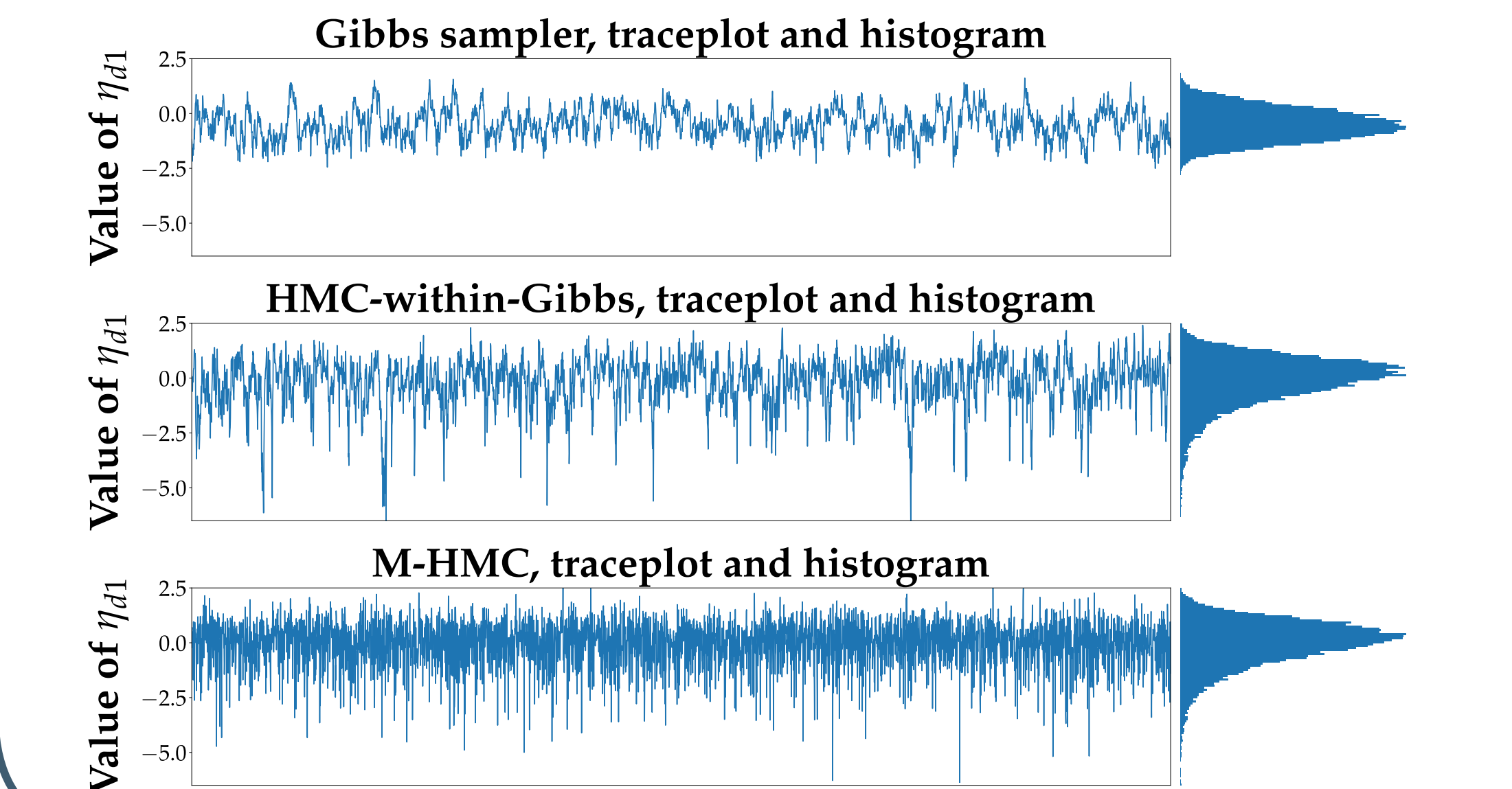1. $\eta_d \sim N(\mu, \Sigma)$
2. For each $n \in \{1, \cdots, N_d\}$:
   (a) Draw topic assignment $z_{d,n}|\eta_d \sim Categ(f(\eta_d))$
   (b) Draw word $w_{d,n}|z_{d,n}, \beta \sim Categ(\beta_{z_{d,n}})$

Target distribution: $p(\eta, z|w; \beta, \mu, \Sigma)$

Methods tested: M-HMC, DHMC, HwG, NwG, Gibbs (Chen et al., 2013)

Results summary:
- DHMC fails; Gibbs occlusionally fails
- M-HMC is 8x more efficient than HwG/NwG
- M-HMC is 60x more efficient than Gibbs

Gibbs sampler, traceplot and histogram

HMC-within-Gibbs, traceplot and histogram

M-HMC, traceplot and histogram

4 Conclusions

- M-HMC evolves discrete and continuous variables in tandem, and is applicable to any distributions with mixed support.
- M-HMC with Laplace momentum is easy to implement, and introduces minimal overhead when compared with existing HMC methods.
- M-HMC is shown to be more efficient than strong baselines on challenging distributions with mixed support.