Beyond Self-Repellent Kernels: History-Driven Target Towards Efficient Nonlinear MCMC on General Graphs

Jie Hu, Yi-Ting Ma, Do Young Eun

Department of Electrical and Computer Engineering
NC State University

ICML 2025
Vancouver, Canada

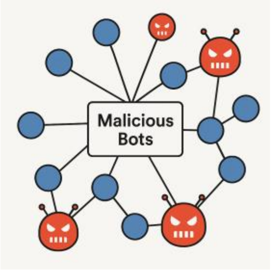
MCMC on General Graphs

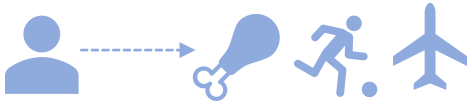
Markov Chain Monte Carlo (MCMC) on General Graphs

- A fundamental tool for understanding graphs, including discrete spaces:
  - E.g., social networks, IoTs, smart grids, biochemical molecules, Ising/Potts models, etc.
- <u>Draw samples</u> from a ***Known*** Distribution (up to a multiplicative constant) $\mu(x) \propto \exp(-H(x)/T)$
- Estimates $E_\mu\{f(X)\} = \sum_{x \in \mathcal{X}} f(x)\mu(x)$ when analyzing entire finite state space is ***infeasible***

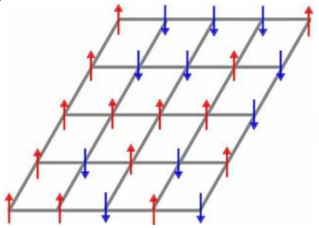
Applications:

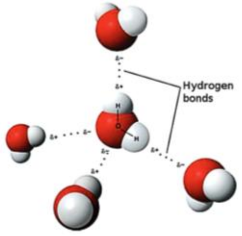
✓ Detect malicious bots & malware spread
✓ Identify key influencers or customer groups
✓ Infer user's preference





✓ Recommendation Systems
✓ Web Crawling / Search Index
✓ Monte Carlo Molecular Modeling
✓ Ising and Potts Models
✓ Energy-based Models





Algorithmic Design of MCMC

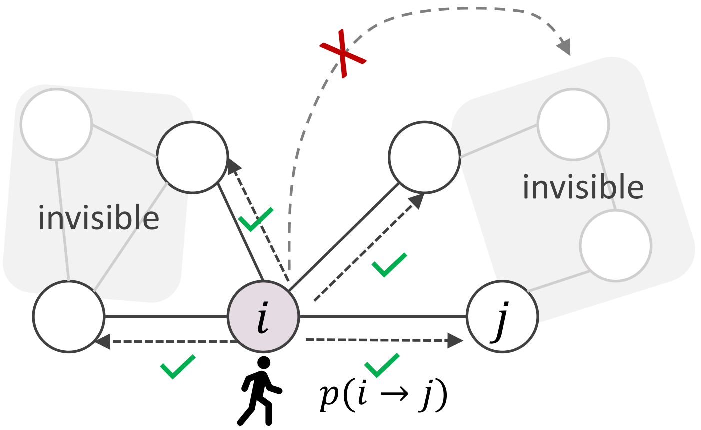
Key Design Criteria for Efficient Graph Samplers:

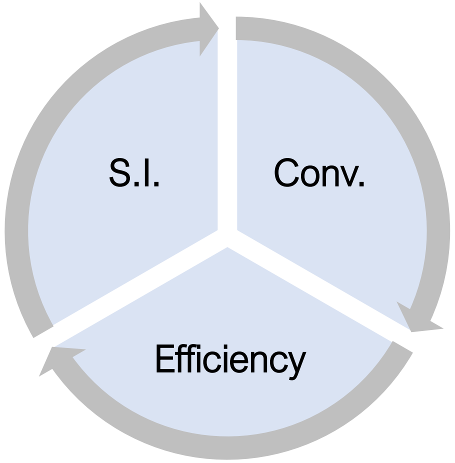
$$\mu(x) = \exp(-H(x)/T)/Z, \text{ where } Z = \int_x \exp(-H(x)/T)$$

1. *Scale Invariant (S.I.)*: Operate w/o global information Z of the graph
2. Robust Theoretical *Convergence (conv.)*: Guarantee convergence to the target distribution

Determine transition probability $p(i \rightarrow j)$ from node i to node j

3. *Efficiency*: Requires *fewer samples* to achieve a similar level of approximation accuracy





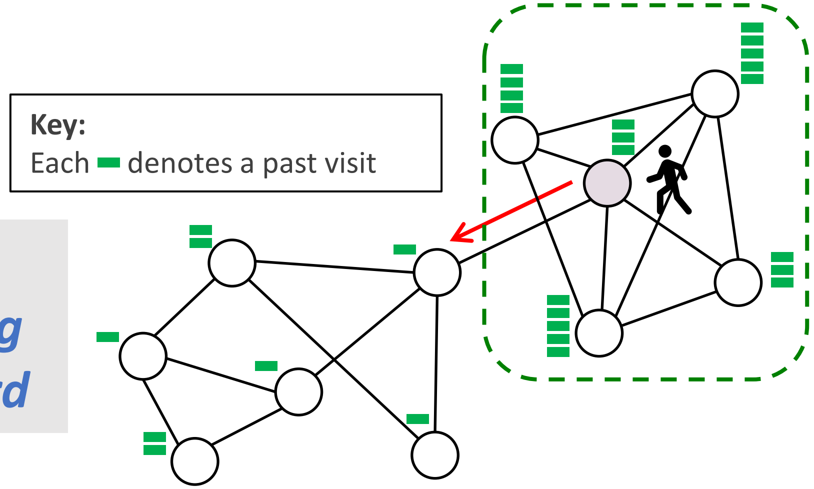
Improving MCMC – Self Interactions

Our recent breakthrough: Self-Repellent Random Walk (SRRW)

- **Concept:** Use the random walker's *history* to influence future transitions
  - Given a time-reversible Markov chain $\boldsymbol{P}$ with target probability distribution $\boldsymbol{\mu}$
  - Based on visit frequency vector $\mathbf{x}$, modify probability from node $i \to j$: $K[\mathbf{x}]_{ij} \propto P_{ij}\left(\frac{x_j}{\mu_j}\right)^{-\alpha}$
  - *'Non-Markov' or 'history-aware' walker*

nonlinear kernel

ICML 2023 Outstanding Paper Award



- Tackle a challenging open problem, MCMC with *self-repellent scheme for the first time*
- Beyond traditional *non-backtracking* approaches which avoid the immediate previous sample

Vishwaraj Doshi, Jie Hu, and Do Young Eun, *"Self Repelling Random Walks on General Graphs – Achieving Minimal Sampling Variance via Non-linear Markov Chains"*, ICML, 2023

Improving MCMC – Self Interactions

Benefits:

- Generated samples still converges to the correct target $\boldsymbol{\mu}$
- Exhibits *S.I.* property: $K[\mathbf{x}]_{ij} \propto P_{ij}\left(\frac{x_j}{\mu_j}\right)^{-\alpha}$ proved to the only form to adjust the kernel P w/o knowing Z
- Achieves *much better performance*

$$\sqrt{n}(\mathbf{x}_n - \boldsymbol{\mu}) \xrightarrow[dist.]{n\to\infty} N\big(\mathbf{0}, \boldsymbol{V}_{\mathbf{x}}(\alpha)\big)$$

and the *near-zero* sampling variance $\boldsymbol{V}_{\mathbf{x}}(\alpha) = O(1/\alpha)$

More efficient than i.i.d sampler under topological constraints!

Vishwaraj Doshi, Jie Hu, and Do Young Eun, *"Self Repelling Random Walks on General Graphs – Achieving Minimal Sampling Variance via Non-linear Markov Chains"*, ICML, 2023

The Catch: Issues Overlooked in SRRW

1. <u>Computational issues</u>: SRRW requires pre-computation of P_{ij} for all j

Standard Metropolis-Hastings

Lightweight & On-Demand

Step 1: Propose one neighbor j w.p. Q_{ij}

Step 2: Calculate acceptance rate

$$A(i,j) = \min\left\{1, \frac{\mu_j Q_{ji}}{\mu_i Q_{ij}}\right\}$$

Cost of acquiring a neighbor's information: $O(1)$

Step 3: Flip a coin to decide the movement

Key Idea: The probability of staying at i ($P_{ii} = 1 - \sum_j P_{ij}$) is an *implicit outcome* of rejection. It is **never pre-computed**.

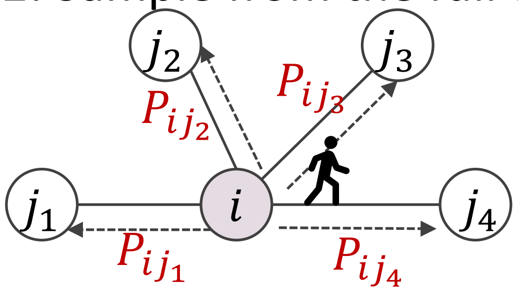
Self-Repellent Random Walk

Heavy & Pre-Computed Transition Probability

Step 1: Compute prob. to each neighbor P_{ij}

(including self-transition P_{ii})

Step 2: Sample from the full distribution and move



$$P_{ii} = 1 - \sum_{k=1}^{4} P_{ij_k}$$

Cost: $O(\deg_i)$

Problem: Need P_{ij} for all j pre-computed, destroying the lightweight nature of MH.
The cost for one sample **scales with the node's degree**, making it extremely slow in dense graphs.

The Catch: Issues Overlooked in SRRW

2. <u>Reversibility</u>: Requires $\boldsymbol{P}$ to be reversible w.r.t. the given target μ (i.e., $\mu_i P_{ij} = \mu_j P_{ji}$)

- A requirement to ensure a well-defined stationary distribution $\boldsymbol{\pi}[\mathbf{x}]$ for the SRRW kernel $\boldsymbol{K}[\mathbf{x}]$
- E.g., $\pi_i[\mathbf{x}]K_{ij}[\mathbf{x}] = \pi_j[\mathbf{x}]K_{ji}[\mathbf{x}], \quad \forall i,j \in \mathcal{V}, \mathbf{x} \in \mathrm{Int}(\boldsymbol{\Sigma})$
- Exclude a whole class of efficient, advanced ***non-reversible*** MCMC samplers

3. <u>Memory Constraints</u>: Dimension of $\mathbf{x}_n$ = the size (#) of state space

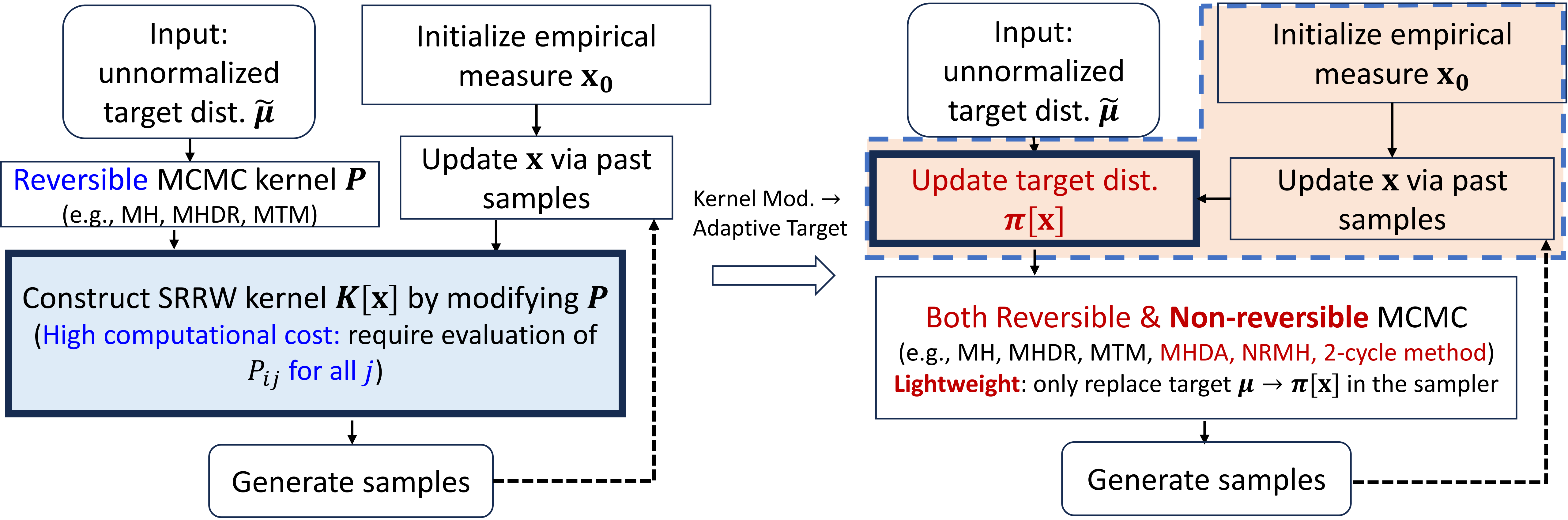
History-Driven Target (HDT) MCMC:

Tackle first two issues of SRRW --- computational costs & time-reversibility

- Only takes $\boldsymbol{O}(\mathbf{1})$ computational cost per sample
- Compatible with ***non-reversible MCMC samplers***
- A heuristic remedy for memory issue --- **L**east **R**ecently **U**sed (LRU) cache scheme

Improvement over SRRW: A Simple Paradigm Shift

Instead of altering the walker's kernel, we *modify the target distribution* *(based on history)*



(a). Self-Repellent Random Walk

(b). History-Driven Target (HDT) Framework

The History-Driven Target (HDT) Framework

- **Our History-Driven Target (HDT) is simple but powerful**
  - The HDT Formula:

$$\pi_i[\mathbf{x}] \propto \underbrace{\mu_i}_{\text{original target}} \left(\frac{x_i}{\mu_i}\right)^{-\alpha}$$

original target → repellence penalty

- Why HDT is a Game-Changer:
  - ***Universal (Bring your own MCMC):*** Works as a "wrapper" for any MCMC method, including the fast non-reversible ones that SRRW cannot use.
  - ***Lightweight:*** Integrates into any sampler by simply replacing the target μ with $\pi[\mathbf{x}]$, preserving the original $O(1)$ cost.

For example, in MHRW, the acceptance rate

$$A_{ij}[\mathbf{x}] = \min\left\{1, \frac{\pi_j[\mathbf{x}]Q_{ji}}{\pi_i[\mathbf{x}]Q_{ij}}\right\}$$

only unnormalized value is needed for calculation

Key Theoretical Guarantees

Three key theoretical findings (*c.f.* Thoerem 3.3, Corollary 3.4, Lemma 3.6)

1. **Unbiased Sampling:** Proven to converge to the correct target distribution

   - $\mathbf{x}_n \xrightarrow[a.s.]{n\to\infty} \boldsymbol{\mu}$, i.e., empirical measure converges to the target distribution almost surely

2. **Near-Zero Variance:** Same $O(1/\alpha)$ variance reduction as SRRW in the CLT

   - $\sqrt{n}(\mathbf{x}_n - \boldsymbol{\mu}) \xrightarrow[dist.]{n\to\infty} N\big(\mathbf{0}, \boldsymbol{V}_{\mathbf{x}}(\alpha)\big)$, where sampling variance $\boldsymbol{V}_{\mathbf{x}}(\alpha) = \frac{1}{2\alpha+1}\boldsymbol{V}^{base}$

3. **Superior Cost-Efficiency:** Provably more efficient than SRRW under same budget

   - $\sqrt{B}(\mathbf{x}_B - \boldsymbol{\mu}) \xrightarrow[a.s.]{B\to\infty} N\big(\mathbf{0}, \boldsymbol{V}_{cost}(\alpha)\big)$, and cost-based sampling variance $\boldsymbol{V}_{cost}^{HDT}(\alpha) \leq_L \frac{2}{\text{avg deg}+1} \cdot \boldsymbol{V}_{cost}^{SRRW}(\alpha)$

Budget of computation B

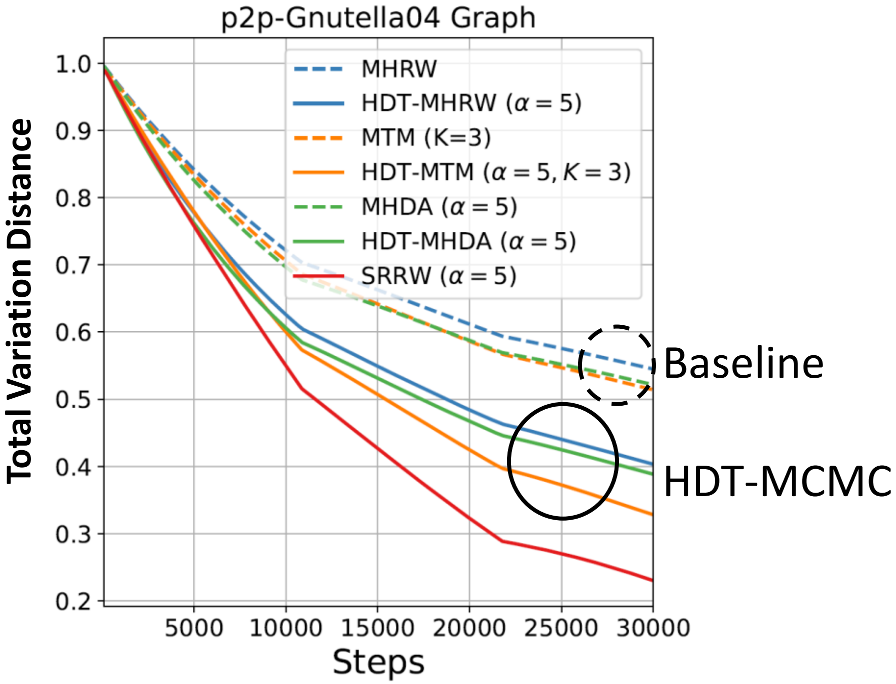
Loewner ordering

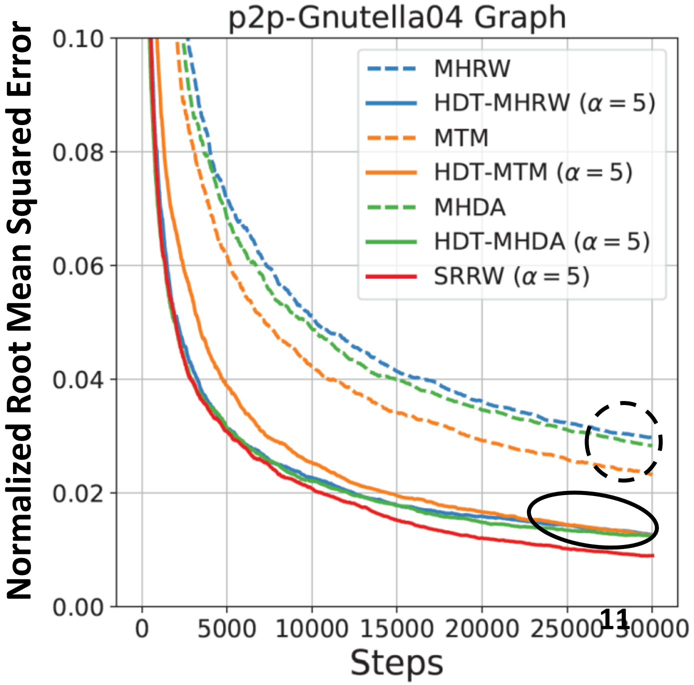
Result 1: HDT Universally Boosts Sampler Performance

HDT-MCMC delivers the best of both worlds for MCMC tasks:

- Draw samples from uniform target on multiple real-world graphs
- Test over multiple baseline MCMC samplers, including
  - Metropolis-Hastings Random Walk (MHRW), Multiple-Try Metropolis (MTM), MH with delayed acceptance (MHDA) non-reversible

Improvement over ***any*** MCMC sampler
Solid lines (HDT version) v.s. dash lines (baseline)



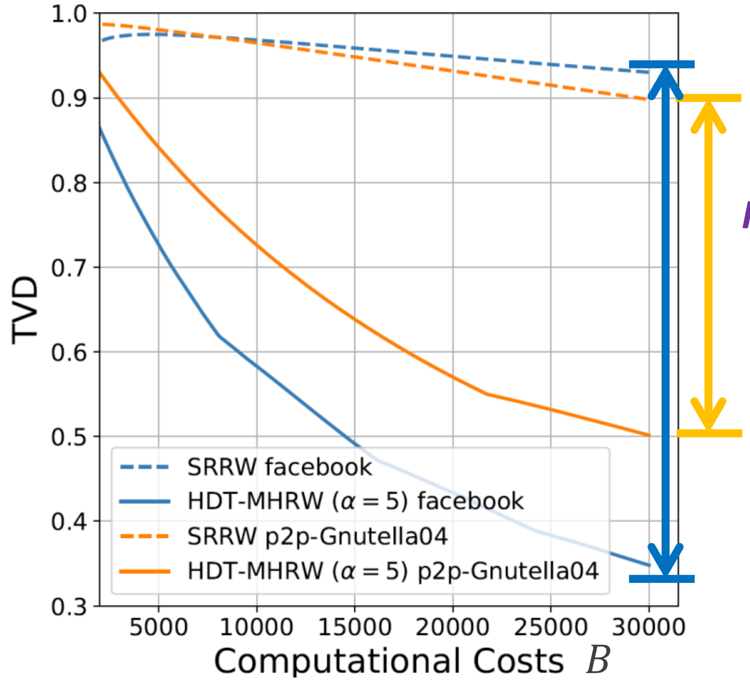


Result 2: HDT is More Cost-Efficient than SRRW

Compare HDT-MHRW and SRRW under a fixed computational budget B

- Lightweight design, HDT-MCMC more cost efficient than SRRW

Average Degree of Graph
Facebook: 43.6
P2p-Gnutella04: 7.4



Improvement more pronounced in the facebook graph

Computational cost per sample at node i:

- HDT-MCMC: 1
- SRRW: $\deg_i$ (degree of node i) due to the pre-computation of all P_{ij} of neighbor j

Denser the graph
$\Rightarrow$ ***larger*** the average neighborhood size
$\Rightarrow$ ***smaller*** covariance

Heuristic Memory Reduction Scheme

Least Recently Used (LRU) cache scheme

- ***Essential idea:*** track only recently visited states, discarding the least-recently used when capacity in cache $\mathcal{C}$ is reached, whose size $|\mathcal{C}| = r|\mathcal{V}|$ (r acts as *the compression ratio*)
- Leverages temporal locality: non-neighboring states do not affect self-repellency
- For a neighbor $j \notin \mathcal{C}$ of current state i, extrapolate its frequency $\hat{x}_j$ via

$$\frac{\hat{x}_j}{\mu_j} = \sum_{k \in (\mathcal{N}(i) \cup \{i\}) \cap \mathcal{C}} \frac{1}{|(\mathcal{N}(i) \cup \{i\}) \cap \mathcal{C}|} \cdot \frac{x_k}{\mu_k}$$

Assume the level of 'deviation' of node j is similar to its neighbors in the cache $\mathcal{C}$

'Deviation' of past visit of node k in the cache $\mathcal{C}$ compared to its target value

Average level of 'deviation' from neighbors that are in the cache $\mathcal{C}$

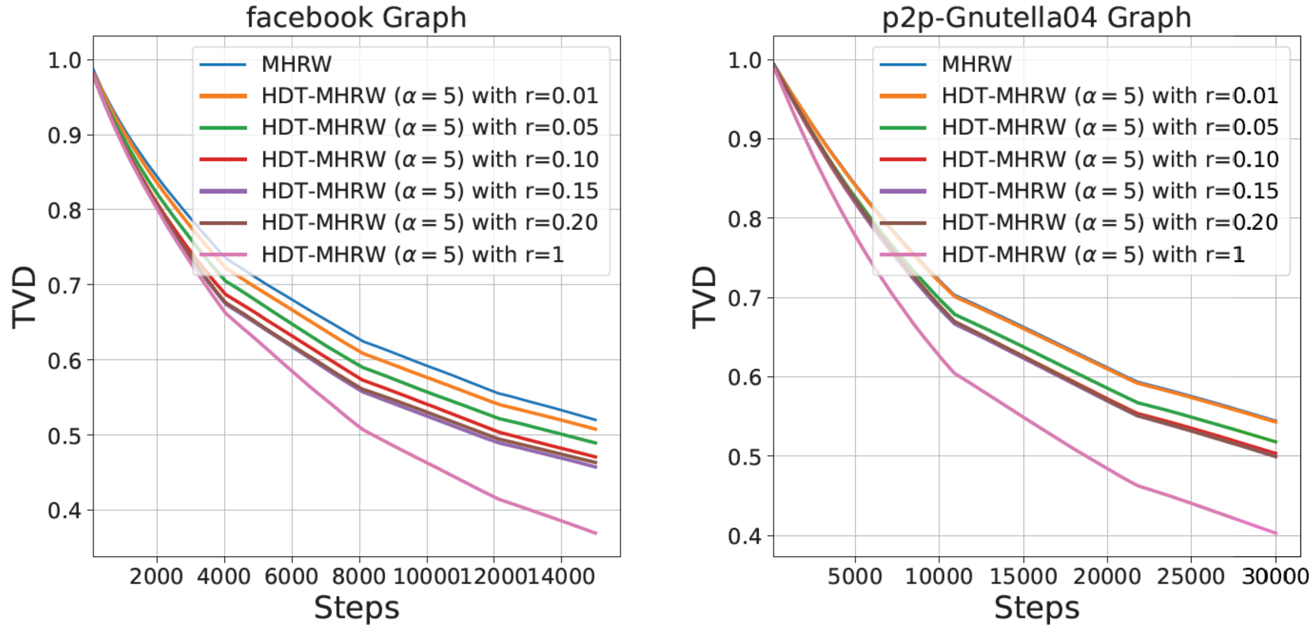
Result 3: HDT is Scalable and Memory-Efficient

Least Recently Used (LRU) cache scheme

- For a neighbor $j \notin \mathcal{C}$ of current state i, extrapolate its frequency via

$$\hat{x}_j = \mu_j \sum_{k \in (\mathcal{N}(i) \cup \{i\}) \cap \mathcal{C}} \frac{1}{|(\mathcal{N}(i) \cup \{i\}) \cap \mathcal{C}|} \cdot \frac{x_k}{\mu_k}$$

- HDT-MHRW w/ LRU robust to the choice of r (compression ratio) leads to **10%** smaller TVD than MHRW with over **90%** memory reduction $(r = 0.1)$



Conclusion

Our previous work SRRW – first to utilize history with theoretical analysis to show near-zero sampling variance

HDT makes history-aware MCMC a practical, powerful, and universal tool:

- ✅ **Paradigm shift** from kernel mod to target mod: Retain *near-zero* variance benefits
- ✅ **Universal and lightweight "wrapper“**: accelerate any MCMC sampler on discrete spaces
- ✅ Provably more **cost-efficient** than the previous state-of-the-art
- ✅ **Scalable to large graphs** via a memory-saving LRU cache scheme

Thank you!

Q&A

Feel free to chat with us at East Exhibition Hall A-B #E-1304 (11 am – 1:30 pm)

Paper link [arXiv preprint]

Backup Slides

SRRW vs. HDT-MCMC: A Comparison

Feature	SRRW (Self-Repellent Random Walk) ICML 2023 Outstanding Paper Award	HDT-MCMC (History-Driven Target MCMC) This work
Core Idea	Tweaks the kernel $K[\mathbf{x}]_{ij} \propto P_{ij}\left(\frac{x_j}{\mu_j}\right)^{-\alpha}$	Tweaks the target $\pi_i[\mathbf{x}] \propto \mu_i\left(\frac{x_i}{\mu_i}\right)^{-\alpha}$
Baseline Chain	Must be time-reversible	Works with advanced **non-reversible** (faster) chains AND reversible ones *Cor. 3.4*
Computational Cost	Higher (needs to evaluate P_{ij} for all neighbors)	**Lower** (simpler to implement with existing samplers)
Ergodicity	√	√ (Thm. 3.3)
Asymptotic Covariance	$\boldsymbol{V}^{SRRW}(\alpha) \sim O(1/\alpha)$	$\boldsymbol{V}^{HDT}(\alpha) \sim O(1/\alpha)$ (Thm. 3.3)
Key Advantage	Groundbreaking performance	**Often better practical speed** due to lower cost & adoption of non-reversible chains†

† Covariance of HDT-MCMC $\approx \frac{1}{avg\ degree}$ of SRRW *under same budget of computation (cf. Lemma 3.6)*

HDT-MCMC: Deterministic analysis

- The closed form of $\boldsymbol{\pi}(\mathbf{x}) = [\pi_i(\mathbf{x})], \forall i \in \mathcal{V}$, is given by

$$\pi_i(\mathbf{x}) = \frac{\mu_i \left(\frac{x_i}{\mu_i}\right)^{-\alpha}}{\sum_k \mu_k \left(\frac{x_k}{\mu_k}\right)^{-\alpha}}$$

$= \omega(\mathbf{x})$

Theorem (Global stability of ODE) For all $\alpha \geq 0$, $\mathbf{x}(0) \in \mathrm{Int}(\Sigma)$, we have

$$\mathbf{x}(t) \longrightarrow \boldsymbol{\mu} \quad \text{as} \quad t \to \infty,$$

where $\boldsymbol{\mu} = [\mu_i] \in \mathrm{Int}(\Sigma)$ is the target stationary distribution, and $\mathbf{x}(t)$ is the solution (trajectory) of the mean-field ODE $\dot{\mathbf{x}}(t) = \boldsymbol{\pi}(\mathbf{x}(t)) - \mathbf{x}(t)$.

- The proof steps are similar to the ODE analysis of SRRW.

HDT-MCMC: Stochastic Analysis

Theorem 1 (Strong Law of Large Number (SLLN) and Central Limit Theorem (CLT)) For all $\alpha \geq 0$, any $\mathbf{x}_0 \in \mathrm{Int}(\Sigma)$, and any $X_0 \in [N]$, we have

$$\mathbf{x}_n \longrightarrow \boldsymbol{\mu} \quad \text{as } n \to \infty, \qquad \textit{almost surely}$$

$$\sqrt{n}(\mathbf{x}_n - \boldsymbol{\mu}) \longrightarrow N\big(\mathbf{0}, \boldsymbol{V}_{\mathbf{x}}(\alpha)\big) \quad as\, n \to \infty, \qquad in\, dist.$$

where $N\big(\mathbf{0}, \boldsymbol{V}(\alpha)\big)$ is a normal distribution with mean $\mathbf{0}$ and covariance $\boldsymbol{V}_{\mathbf{x}}(\alpha)$, given by

$$\boldsymbol{V}_{\mathbf{x}}(\alpha) = \frac{1}{2\alpha + 1}\boldsymbol{V}^{base} = O(1/\alpha)$$

Corollary 2 (Preserved Efficiency Ordering)

Suppose two MCMC samplers S_1 and S_2 converge to $\boldsymbol{\mu}$ with limiting covariances $\boldsymbol{V}^{S_1}$ and $\boldsymbol{V}^{S_2}$ satisfying

$$\boldsymbol{V}^{S_1} <_L \boldsymbol{V}^{S_2},$$

Meaning that sampler S_1 is more efficient than sampler S_2. Applying HDT framework to both, yielding $V^{S_1-HDT}(\alpha)$ and $V^{S_2-HDT}(\alpha)$, preserves the ordering:

$$V^{S_1-HDT}(\alpha) <_L V^{S_2-HDT}(\alpha), \forall \alpha \geq 0.$$

Any known covariance orderings between reversible and non-reversible samplers carry over to HDT-MCMC, whereas SRRW cannot accommodate non-reversible Markov chains.

HDT-MCMC: Cost-Related Analysis

Let a_i (resp. b_i) $\in (0, \infty)$ be the ***computational cost*** of the i-th sample in HDT-MCMC (resp. SRRW). Define:

$$T^{HDT}(B) := \max\{k \mid a_1 + a_2 + \cdots + a_k \leq B\}$$
$$T^{SRRW}(B) := \max\{k' \mid b_1 + b_2 + \cdots + b_{k'} \leq B\}$$

the number of samples that HDT-MCMC (resp. SRRW) can generate *before hitting the total budget B.*

Average computational cost of HDT, SRRW

Theorem 7 (Cost-Based CLT)
Suppose that as $B \to \infty$,

$$B/T^{HDT}(B) \to C^{HDT}, \qquad B/T^{SRRW}(B) \to C^{SRRW} \quad a.s.$$

Then, we have

$$\sqrt{B}\left(\mathbf{x}_{T^{HDT}(B)} - \boldsymbol{\mu}\right) \longrightarrow N\left(\mathbf{0}, C^{HDT}\mathbf{V}_{\mathbf{x}}^{HDT}(\alpha)\right) \quad as\ n \to \infty, \quad in\ dist.$$
$$\sqrt{B}\left(\mathbf{x}_{T^{SRRW}(B)} - \boldsymbol{\mu}\right) \longrightarrow N\left(\mathbf{0}, C^{SRRW}\mathbf{V}_{\mathbf{x}}^{SRRW}(\alpha)\right) \quad as\ n \to \infty, \quad in\ dist.$$

HDT-MCMC: Cost-Related Analysis

Theorem 3 (Cost-based CLT)
Suppose that as $B \to \infty$,

$$B/T^{HDT}(B) \to C^{HDT}, \qquad B/T^{SRRW}(B) \to C^{SRRW} \quad a.s.$$

Then, we have

$$\sqrt{B}\left(\mathbf{x}_{T^{HDT}(B)} - \boldsymbol{\mu}\right) \longrightarrow N\left(\mathbf{0}, C^{HDT}\mathbf{V}_{\mathbf{x}}^{HDT}(\alpha)\right) \qquad as\ n \to \infty, \qquad in\ dist.$$

$$\sqrt{B}\left(\mathbf{x}_{T^{SRRW}(B)} - \boldsymbol{\mu}\right) \longrightarrow N\left(\mathbf{0}, C^{SRRW}\mathbf{V}_{\mathbf{x}}^{SRRW}(\alpha)\right) \quad as\ n \to \infty, \qquad in\ dist.$$

Lemma 8 (Ordering of cost-based covariances between HDT-MCMC and SRRW)

$$C^{HDT}\mathbf{V}_{\mathbf{x}}^{HDT}(\alpha) <_L \frac{2}{E_{i\sim\boldsymbol{\mu}}[\mathcal{N}(i)] + 1} C^{SRRW}\mathbf{V}_{\mathbf{x}}^{SRRW}(\alpha)$$

- Cost-based covariance of HDT-MCMC **at least a factor of $\frac{2}{E_{i\sim\mu}[\mathcal{N}(i)]+1}$ times smaller** than that of SRRW for every given α, suggesting a universal advantage.
- ***Denser*** the graph, ***larger*** the average neighborhood size $E_{i\sim\mu}[\mathcal{N}(i)]$, ***smaller*** covariance