Massachusetts Institute of Technology

LIDS

Many Processors, Little Time: MCMC for Partitions via Optimal Transport Couplings

Brian L. Trippe

Postdoctoral Research Fellow

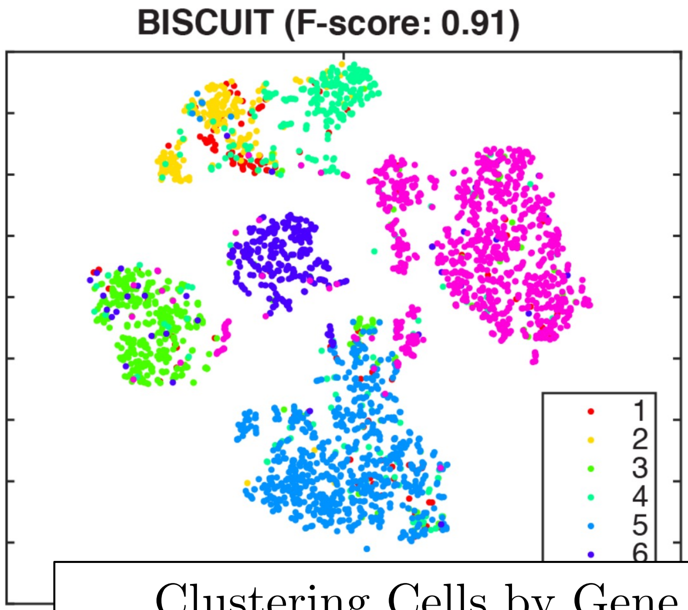
Columbia University Department of Statistics

Tin D. Nguyen Tamara Broderick

Unsupervised Learning as Inference on Partitions

Example: Cluster cells based on gene expression [1]

- How large are the clusters (cell types)?
- Which cells are of the same type?



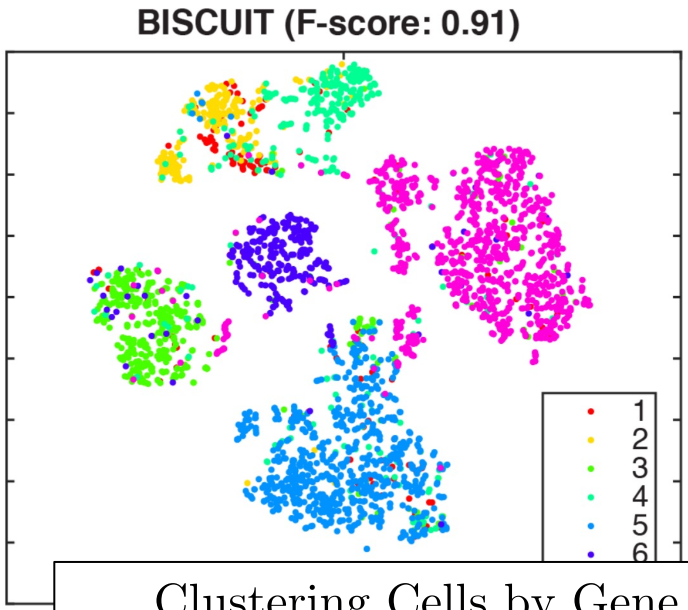
Clustering Cells by Gene Expression (Prabhakaran, 2016)

Unsupervised Learning as Inference on Partitions

Example: Cluster cells based on gene expression [1]

- How large are the clusters (cell types)?
- Which cells are of the same type?

Intractability of Bayesian posterior → MCMC

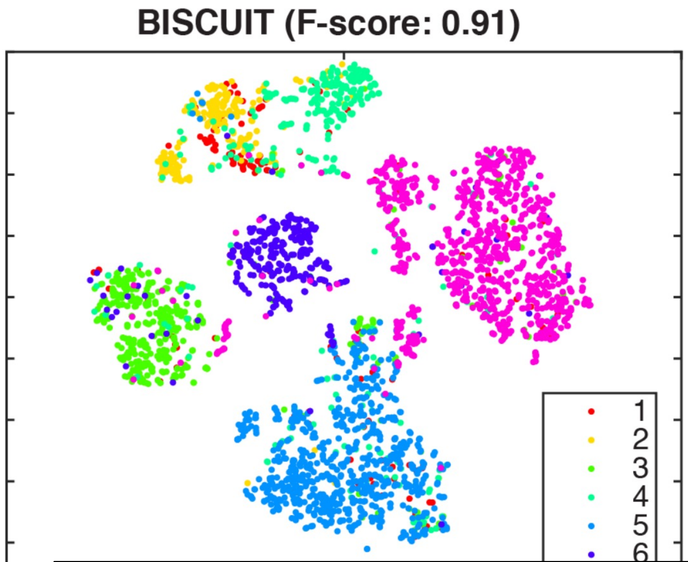


Clustering Cells by Gene Expression (Prabhakaran, 2016)

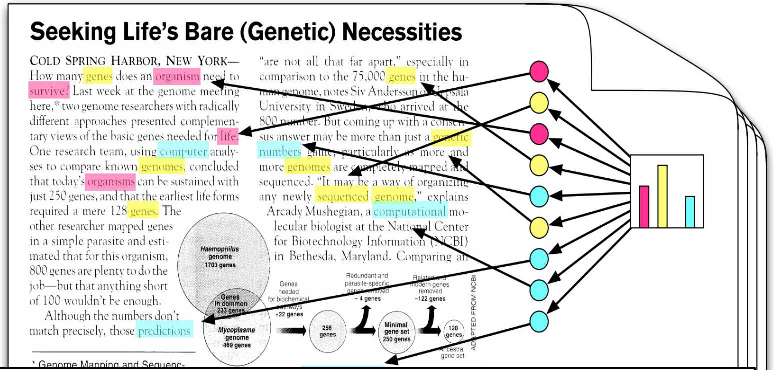
Unsupervised Learning as Inference on Partitions

Example: Cluster cells based on gene expression [1]
- How large are the clusters (cell types)?
- Which cells are of the same type?

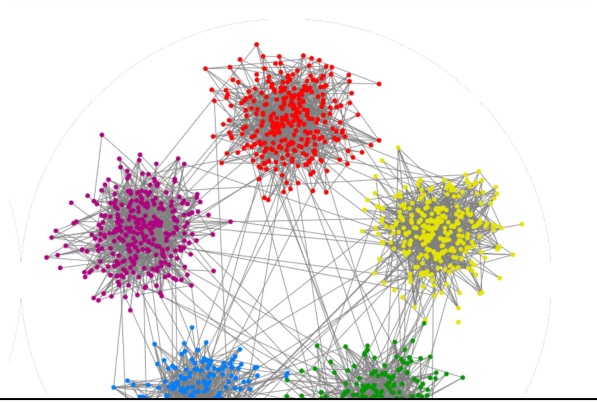
Intractability of Bayesian posterior → MCMC



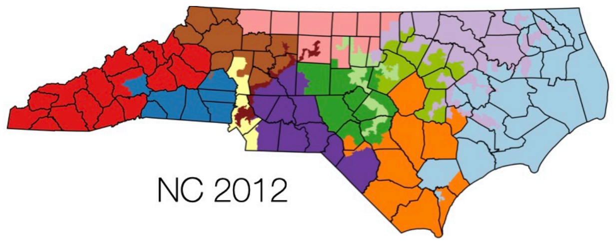
Clustering Cells by Gene Expression (Prabhakaran, 2016)



Topic Modeling (Blei, 2012)

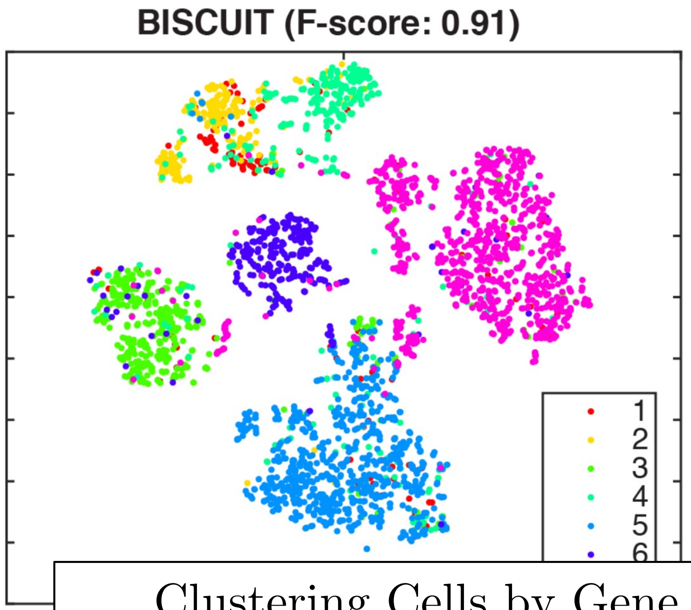


Community Detection (Abbe, 2017)



Computional Redistricting (MIT Tech. Review)

Unsupervised Learning as Inference on Partitions

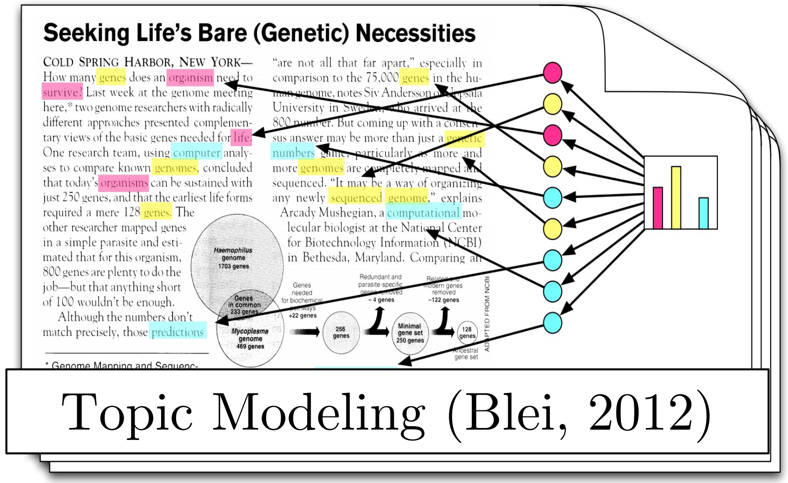


Clustering Cells by Gene Expression (Prabhakaran, 2016)

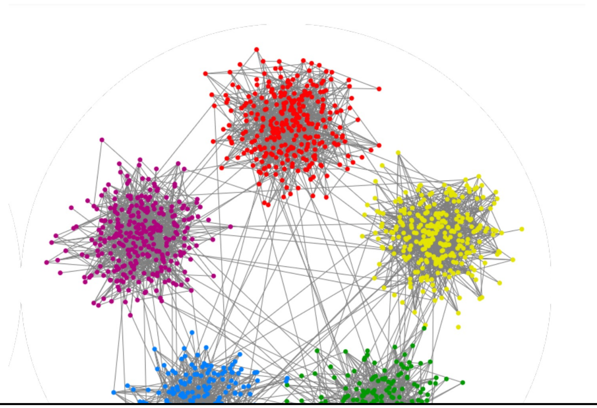
Example: Cluster cells based on gene expression [1]

- How large are the clusters (cell types)?
- Which cells are of the same type?

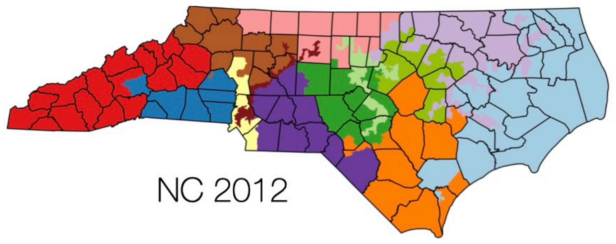
Intractability of Bayesian posterior → MCMC



Topic Modeling (Blei, 2012)



Community Detection (Abbe, 2017)



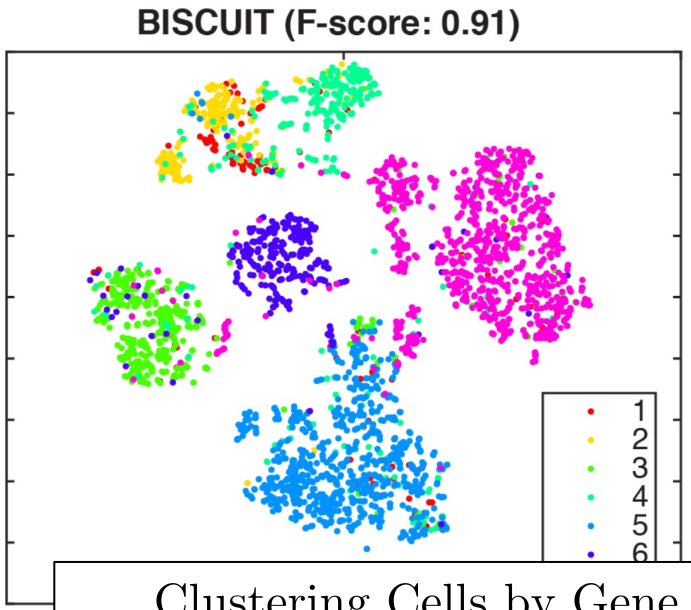
Computional Redistricting (MIT Tech. Review)

Challenge: MCMC takes a long time!

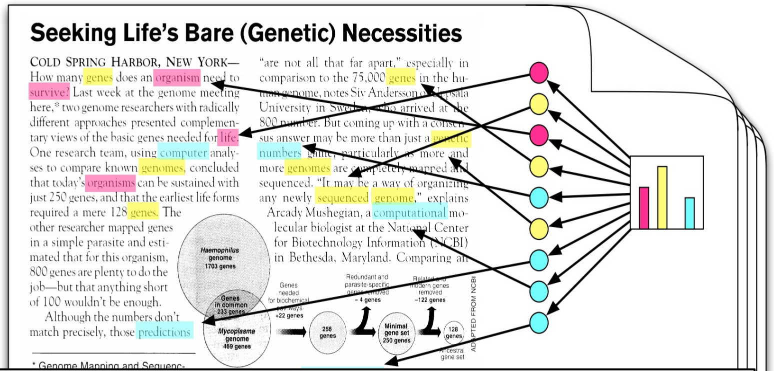
Unsupervised Learning as Inference on Partitions

Example: Cluster cells based on gene expression [1]
- How large are the clusters (cell types)?
- Which cells are of the same type?

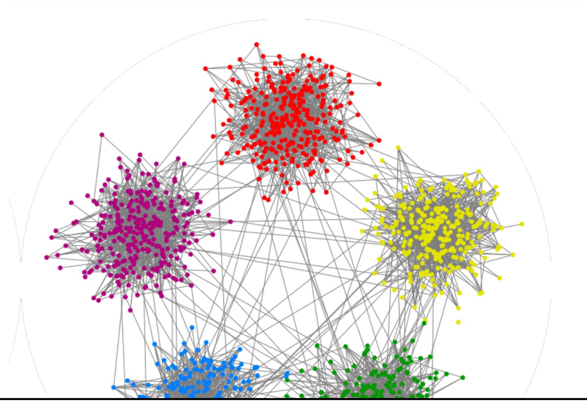
Intractability of Bayesian posterior → MCMC



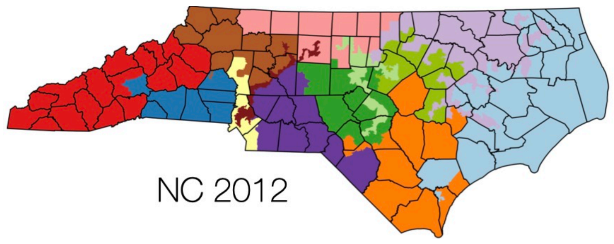
Clustering Cells by Gene Expression (Prabhakaran, 2016)



Topic Modeling (Blei, 2012)



Community Detection (Abbe, 2017)



Computional Redistricting (MIT Tech. Review)

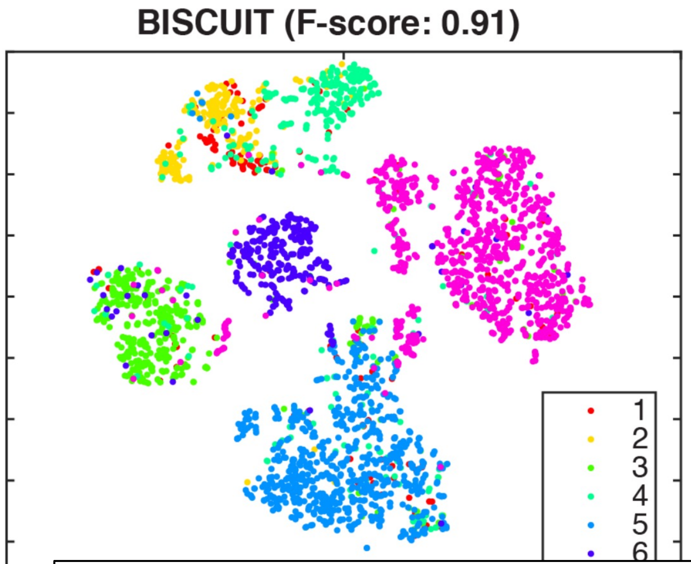
Challenge: MCMC takes a long time!

Goal: Use computational parallelism to accelerate MCMC

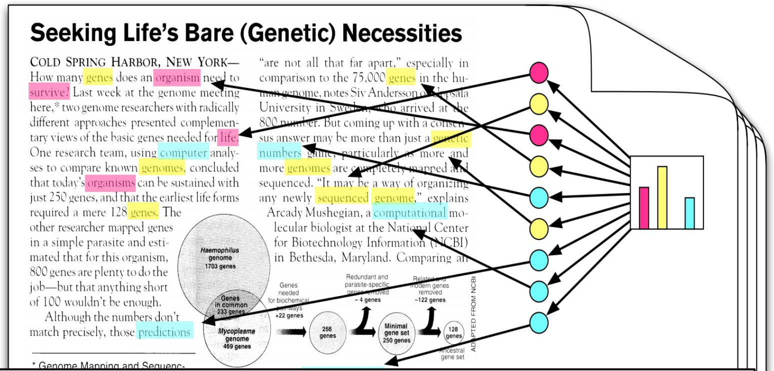
Unsupervised Learning as Inference on Partitions

Example: Cluster cells based on gene expression [1]
- How large are the clusters (cell types)?
- Which cells are of the same type?

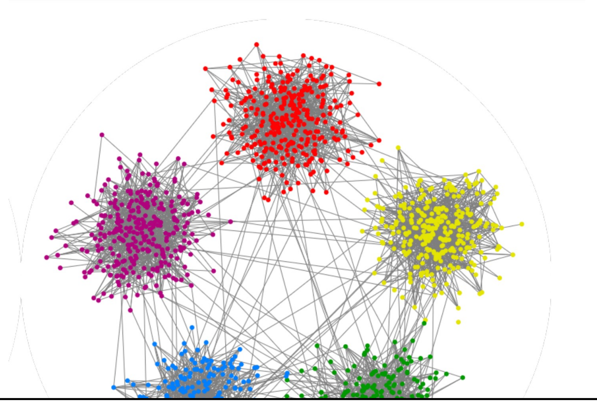
Intractability of Bayesian posterior → MCMC



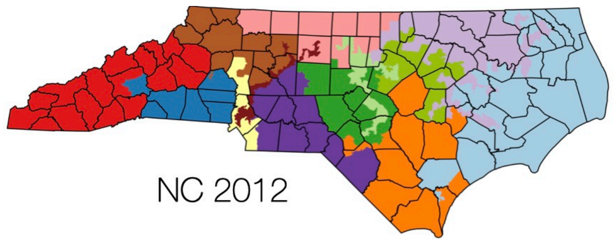
Clustering Cells by Gene Expression (Prabhakaran, 2016)



Topic Modeling (Blei, 2012)



Community Detection (Abbe, 2017)



Computional Redistricting (MIT Tech. Review)

Challenge: MCMC takes a long time!

Goal: Use computational parallelism to accelerate MCMC

This work: Optimal transport couplings make this possible

Roadmap

- Parallelizing MCMC with Couplings:
  - Background & Notation
  - The Label Switching Problem
- We Frame Gibbs Sampling as Markov Chain on Partitions
- Our Optimal Transport Coupling
- Big-O Analysis Demonstrates Fast Computation
- Improved Estimation Error and Intervals with OTC over Naïve Parallelism in Practice

Parallel MCMC with Couplings, Background & Notation

Set-up & challenge of burn-in bias:

- Want to compute $H^* = \int h(X) p_X(X) dX$

Parallel MCMC with Couplings, Background & Notation

Set-up & challenge of burn-in bias:

- Want to compute $H^* = \int h(X) p_X(X) dX$
- Choose Markov chain: $X_0, X_1, \ldots \rightsquigarrow p_X$

Parallel MCMC with Couplings, Background & Notation

Set-up & challenge of burn-in bias:

- Want to compute $H^* = \int h(X)p_X(X)dX$
- Choose Markov chain: $X_0, X_1, \ldots \rightsquigarrow p_X$
- Problem: for any $T < \infty$, $\mathbb{E}[h(X_T)] \neq H^*$

Parallel MCMC with Couplings, Background & Notation

Set-up & challenge of burn-in bias:

- Want to compute $H^* = \int h(X) p_X(X) dX$
- Choose <u>Markov chain</u>: $X_0, X_1, \ldots \rightsquigarrow p_X$
- Problem: for any $T < \infty$, $\mathbb{E}[h(X_T)] \neq H^*$
- "Naïve parallelism" gives little help!

Parallel MCMC with Couplings, Background & Notation

Set-up & challenge of burn-in bias:

- Want to compute $H^* = \int h(X) p_X(X) dX$
- Choose <u>Markov chain</u>: $X_0, X_1, \ldots \rightsquigarrow p_X$
- Problem: for any $T < \infty$, $\mathbb{E}[h(X_T)] \neq H^*$
- "Naïve parallelism" gives little help!

Unbiased MCMC Set-Up (simplified)[†]

Use "coupled chain" $(Y_0, Y_1, \ldots)$ where:

1. $Y_t \backsim X_t$
2. $Y_{t-1} = X_t$ for $\mathrm{t} > \tau$ ("meeting time")

[†] Jacob et al. "Unbiased Markov chain Monte Carlo methods with couplings." 2020

Parallel MCMC with Couplings, Background & Notation

Set-up & challenge of burn-in bias:

- Want to compute $H^* = \int h(X) p_X(X) dX$
- Choose <u>Markov chain</u>: $X_0, X_1, \ldots \rightsquigarrow p_X$
- Problem: for any $T < \infty$, $\mathbb{E}[h(X_T)] \neq H^*$
- "Naïve parallelism" gives little help!

Unbiased MCMC Set-Up (simplified)[†]

Use "coupled chain" $(Y_0, Y_1, \ldots)$ where:

1. $Y_t \backsim X_t$

2. $Y_{t-1} = X_t$ for $\mathrm{t} > \tau$ ("meeting time")

Then:

$$\mathbb{E}\Big[\underbrace{h(X_{T'}) + \textstyle\sum_{t=T'+1}^{\tau} h(X_t) - h(Y_{t-1})}_{\text{unbiased estimate}}\Big] = H^*$$

[†] Jacob et al. "Unbiased Markov chain Monte Carlo methods with couplings." 2020

Parallel MCMC with Couplings, Background & Notation

Set-up & challenge of burn-in bias:

- Want to compute $H^* = \int h(X) p_X(X) dX$
- Choose <u>Markov chain</u>: $X_0, X_1, \ldots \rightsquigarrow p_X$
- Problem: for any $T < \infty$, $\mathbb{E}[h(X_T)] \neq H^*$
- "Naïve parallelism" gives little help!

Unbiased MCMC Set-Up (simplified)[†]

Use "coupled chain" $(Y_0, Y_1, \ldots)$ where:

1. $Y_t \backsim X_t$

2. $Y_{t-1} = X_t$ for $\mathrm{t} > \tau$ ("meeting time")

Then:

$$\mathbb{E}\Big[\underbrace{h(X_{T'}) + \textstyle\sum_{t=T'+1}^{\tau} h(X_t) - h(Y_{t-1})}_{\text{unbiased estimate}}\Big] = H^*$$

- Reduce error <u>with parallelism</u>

[†] Jacob et al. "Unbiased Markov chain Monte Carlo methods with couplings." 2020

Parallel MCMC with Couplings, Background & Notation

Set-up & challenge of burn-in bias:

- Want to compute $H^* = \int h(X) p_X(X) dX$
- Choose <u>Markov chain</u>: $X_0, X_1, \ldots \rightsquigarrow p_X$
- Problem: for any $T < \infty$, $\mathbb{E}[h(X_T)] \neq H^*$
- "Naïve parallelism" gives little help!

$$H^* = \lim_{T\to\infty} \mathbb{E}[h(X_T)]$$

Unbiased MCMC Set-Up (simplified)[†]

Use "coupled chain" $(Y_0, Y_1, \ldots)$ where:

1. $Y_t \backsim X_t$

2. $Y_{t-1} = X_t$ for $\mathrm{t} > \tau$ ("meeting time")

Then:

$$\mathbb{E}\Big[\underbrace{h(X_{T'}) + \textstyle\sum_{t=T'+1}^{\tau} h(X_t) - h(Y_{t-1})}_{\text{unbiased estimate}}\Big] = H^*$$

- Reduce error <u>with parallelism</u>

[†] Jacob et al. "Unbiased Markov chain Monte Carlo methods with couplings." 2020

Parallel MCMC with Couplings, Background & Notation

Set-up & challenge of burn-in bias:

- Want to compute $H^* = \int h(X) p_X(X) dX$
- Choose <u>Markov chain</u>: $X_0, X_1, \ldots \rightsquigarrow p_X$
- Problem: for any $T < \infty$, $\mathbb{E}[h(X_T)] \neq H^*$
- "Naïve parallelism" gives little help!

$$H^* = \lim_{T\to\infty} \mathbb{E}[h(X_T)]$$
$$= \lim_{T\to\infty} \mathbb{E}\left[h(X_{T'}) + \overbrace{\sum_{t=T'+1}^{T} h(X_t) - h(X_{t-1})}^{\text{Telescopic sum}}\right]$$

Unbiased MCMC Set-Up (simplified)[†]

Use "coupled chain" $(Y_0, Y_1, \ldots)$ where:

1. $Y_t \backsim X_t$

2. $Y_{t-1} = X_t$ for $\mathrm{t} > \tau$ ("meeting time")

Then:

$$\mathbb{E}\Big[\underbrace{h(X_{T'}) + \textstyle\sum_{t=T'+1}^{\tau} h(X_t) - h(Y_{t-1})}_{\text{unbiased estimate}}\Big] = H^*$$

- Reduce error <u>with parallelism</u>

[†] Jacob et al. "Unbiased Markov chain Monte Carlo methods with couplings." 2020

Parallel MCMC with Couplings, Background & Notation

Set-up & challenge of burn-in bias:

- Want to compute $H^* = \int h(X) p_X(X) dX$
- Choose Markov chain: $X_0, X_1, \ldots \rightsquigarrow p_X$
- Problem: for any $T < \infty$, $\mathbb{E}[h(X_T)] \neq H^*$
- "Naïve parallelism" gives little help!

$$H^* = \lim_{T\to\infty} \mathbb{E}[h(X_T)]$$
$$= \lim_{T\to\infty} \mathbb{E}\left[h(X_{T'}) + \sum_{t=T'+1}^{T} h(X_t) - h(X_{t-1})\right]$$

Unbiased MCMC Set-Up (simplified)[†]

Use "coupled chain" $(Y_0, Y_1, \ldots)$ where:

1. $Y_t \backsim X_t$

2. $Y_{t-1} = X_t$ for $\mathrm{t} > \tau$ ("meeting time")

Then:

$$\mathbb{E}\Big[\underbrace{h(X_{T'}) + \textstyle\sum_{t=T'+1}^{\tau} h(X_t) - h(Y_{t-1})}_{\text{unbiased estimate}}\Big] = H^*$$

- Reduce error with parallelism

[†] Jacob et al. "Unbiased Markov chain Monte Carlo methods with couplings." 2020

Parallel MCMC with Couplings, Background & Notation

Set-up & challenge of burn-in bias:

- Want to compute $H^* = \int h(X) p_X(X) dX$
- Choose Markov chain: $X_0, X_1, \dots \rightsquigarrow p_X$
- Problem: for any $T < \infty$, $\mathbb{E}[h(X_T)] \neq H^*$
- "Naïve parallelism" gives little help!

$$H^* = \lim_{T\to\infty} \mathbb{E}[h(X_T)]$$
$$= \lim_{T\to\infty} \mathbb{E}\left[h(X_{T'}) + \sum_{t=T'+1}^{T} h(X_t) - \boxed{h(Y_{t-1})}\right] \longleftarrow \boxed{Y_{t-1} \backsim X_{t-1}}$$

Unbiased MCMC Set-Up (simplified)[†]

Use "coupled chain" $(Y_0, Y_1, \dots)$ where:

1. $Y_t \backsim X_t$
2. $Y_{t-1} = X_t$ for $t > \tau$ ("meeting time")

Then:

$$\mathbb{E}\Big[\underbrace{h(X_{T'}) + \textstyle\sum_{t=T'+1}^{\tau} h(X_t) - h(Y_{t-1})}_{\text{unbiased estimate}}\Big] = H^*$$

- Reduce error with parallelism

[†] Jacob et al. "Unbiased Markov chain Monte Carlo methods with couplings." 2020

Parallel MCMC with Couplings, Background & Notation

Set-up & challenge of burn-in bias:

- Want to compute $H^* = \int h(X) p_X(X) dX$
- Choose <u>Markov chain</u>: $X_0, X_1, \ldots \rightsquigarrow p_X$
- Problem: for any $T < \infty$, $\mathbb{E}[h(X_T)] \neq H^*$
- "Naïve parallelism" gives little help!

Unbiased MCMC Set-Up (simplified)[†]

Use "coupled chain" $(Y_0, Y_1, \ldots)$ where:

1. $Y_t \backsim X_t$
2. $Y_{t-1} = X_t$ for $t > \tau$ ("meeting time")

Then:

$$\mathbb{E}\Big[\underbrace{h(X_{T'}) + \textstyle\sum_{t=T'+1}^{\tau} h(X_t) - h(Y_{t-1})}_{\text{unbiased estimate}}\Big] = H^*$$

- Reduce error <u>with parallelism</u>

$$H^* = \lim_{T\to\infty} \mathbb{E}[h(X_T)]$$

$$= \lim_{T\to\infty} \mathbb{E}\left[h(X_{T'}) + \sum_{t=T'+1}^{T} h(X_t) - h(Y_{t-1})\right] \leftarrow \boxed{Y_{t-1} \backsim X_{t-1}}$$

$$= \mathbb{E}\left[\mathrm{h}(\mathrm{X}_{\mathrm{T}'}) + \sum_{t=T'+1}^{\boxed{\tau}} h(X_t) - h(Y_{t-1})\right] \leftarrow \boxed{\text{Only finitely many non-zero terms.}}$$

[†] Jacob et al. "Unbiased Markov chain Monte Carlo methods with couplings." 2020

Parallel MCMC with Couplings, Background & Notation

Set-up & challenge of burn-in bias:

- Want to compute $H^* = \int h(X) p_X(X) dX$
- Choose <u>Markov chain</u>: $X_0, X_1, \ldots \rightsquigarrow p_X$
- Problem: for any $T < \infty$, $\mathbb{E}[h(X_T)] \neq H^*$
- "Naïve parallelism" gives little help!

Unbiased MCMC Set-Up (simplified)[†]

Use "coupled chain" $(Y_0, Y_1, \ldots)$ where:

1. $Y_t \backsim X_t$

2. $Y_{t-1} = X_t$ for $\mathrm{t} > \tau$ ("meeting time")

Then:

$$\mathbb{E}\Big[\underbrace{h(X_{T'}) + \textstyle\sum_{t=T'+1}^{\tau} h(X_t) - h(Y_{t-1})}_{\text{unbiased estimate}}\Big] = H^*$$

- Reduce error <u>with parallelism</u>

$$H^* = \lim_{T\to\infty} \mathbb{E}[h(X_T)]$$

$$= \lim_{T\to\infty} \mathbb{E}\left[h(X_{T'}) + \sum_{t=T'+1}^{T} h(X_t) - h(Y_{t-1})\right]$$

$$= \mathbb{E}\left[\mathrm{h}(\mathrm{X}_{\mathrm{T}'}) + \sum_{t=T'+1}^{\tau} h(X_t) - h(Y_{t-1})\right]$$

$\boxed{Y_{t-1} \backsim X_{t-1}}$

Only finitely many non-zero terms.

How do we apply this to clustering problems?

[†] Jacob et al. "Unbiased Markov chain Monte Carlo methods with couplings." 2020

Parallel MCMC with Couplings, Methodological Choices

Unbiased MCMC Set-Up

Use coupled chains such that

1. $Y_t \backsim X_t$
2. $Y_{t-1} = X_t$ for $\mathrm{t} > \tau$

Choices for Clustering Applications:

- Transition kernel for X_t

Parallel MCMC with Couplings, Methodological Choices

Unbiased MCMC Set-Up

Use coupled chains such that

1. $Y_t \backsim X_t$

2. $Y_{t-1} = X_t$ for $\mathrm{t} > \tau$

Choices for Clustering Applications:

- Transition kernel for X_t → Gibbs

Parallel MCMC with Couplings, Methodological Choices

Unbiased MCMC Set-Up

Use coupled chains such that

1. $Y_t \backsim X_t$
2. $Y_{t-1} = X_t$ for $\mathrm{t} > \tau$

Choices for Clustering Applications:

- Transition kernel for X_t → Gibbs
- Coupling that meets quickly

Parallel MCMC with Couplings, Methodological Choices

Unbiased MCMC Set-Up

Use coupled chains such that

1. $Y_t \backsim X_t$
2. $Y_{t-1} = X_t$ for $t > \tau$

Estimate (1 per processor)

$$\underbrace{h(X_{T'})}_{\text{Usual MCMC estimate}} + \underbrace{\sum_{t=T'+1}^{\tau} h(X_t) - h(Y_{t-1})}_{\text{Bias correction}}$$

Choices for Clustering Applications:

- Transition kernel for $X_t \rightarrow$ Gibbs
- Coupling that meets quickly

Parallel MCMC with Couplings, Methodological Choices

Unbiased MCMC Set-Up

Use coupled chains such that

1. $Y_t \backsim X_t$

2. $Y_{t-1} = X_t$ for $t > \tau$

Estimate (1 per processor)

$$\underbrace{h(X_{T'})}_{\text{Usual MCMC estimate}} + \underbrace{\textstyle\sum_{t=T'+1}^{\tau} h(X_t) - h(Y_{t-1})}_{\text{Bias correction}}$$

Choices for Clustering Applications:

- Transition kernel for X_t → Gibbs
- Coupling that meets quickly
  - large τ → high variance

Parallel MCMC with Couplings, Methodological Choices

Unbiased MCMC Set-Up

Use coupled chains such that

1. $Y_t \backsim X_t$
2. $Y_{t-1} = X_t$ for $t > \tau$

Estimate (1 per processor)

$$\underbrace{h(X_{T'})}_{\text{Usual MCMC estimate}} + \underbrace{\sum_{t=T'+1}^{\tau} h(X_t) - h(Y_{t-1})}_{\text{Bias correction}}$$

Choices for Clustering Applications:

- Transition kernel for X_t → Gibbs
- Coupling that meets quickly
  - large τ → high variance
  - not addressed by existing work!

Parallel MCMC with Couplings, Methodological Choices

Unbiased MCMC Set-Up

Use coupled chains such that

1. $Y_t \backsim X_t$
2. $Y_{t-1} = X_t$ for t > τ

Estimate (1 per processor)

$$\underbrace{h(X_{T'})}_{\text{Usual MCMC estimate}} + \underbrace{\sum_{t=T'+1}^{\tau} h(X_t) - h(Y_{t-1})}_{\text{Bias correction}}$$

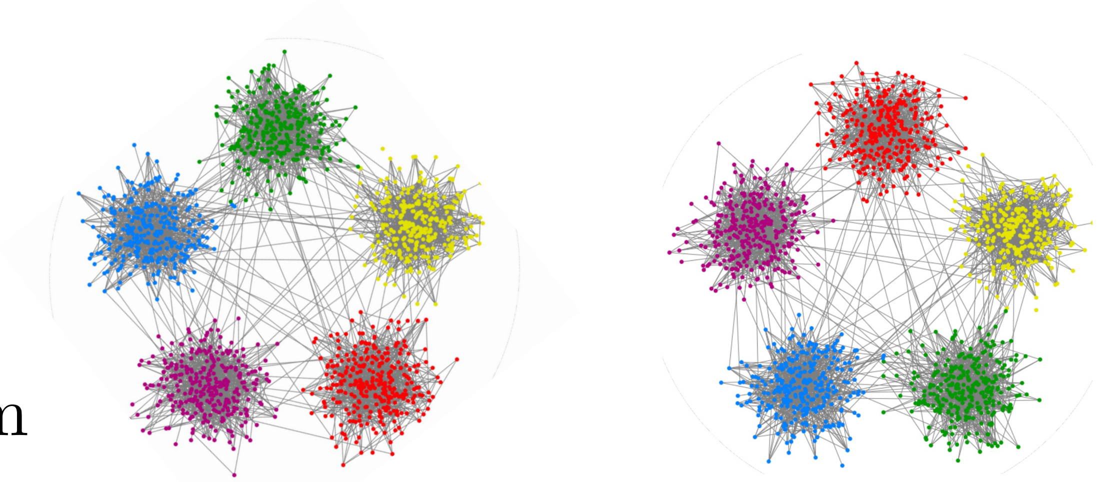
Choices for Clustering Applications:

- Transition kernel for X_t → Gibbs
- Coupling that meets quickly
  - large τ → high variance
  - not addressed by existing work!

Challenge: the "label-switching" problem

Equivalent re-labelings impede mixing

→ slow meeting



Labeling 1 **Labeling 2**

Parallel MCMC with Couplings, Methodological Choices

Unbiased MCMC Set-Up

Use coupled chains such that

1. $Y_t \backsim X_t$
2. $Y_{t-1} = X_t$ for t $> \tau$

Estimate (1 per processor)

$$\underbrace{h(X_{T'})}_{\text{Usual MCMC estimate}} + \underbrace{\sum_{t=T'+1}^{\tau} h(X_t) - h(Y_{t-1})}_{\text{Bias correction}}$$

Choices for Clustering Applications:

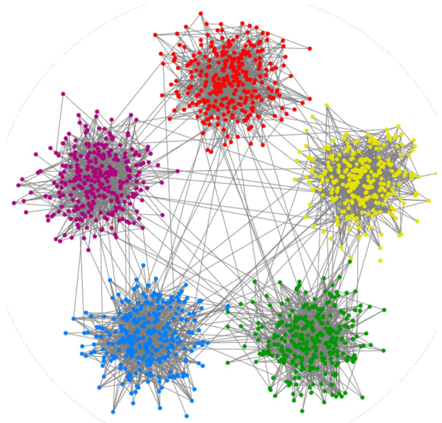
- Transition kernel for X_t → Gibbs
- Coupling that meets quickly
  - large τ → high variance
  - not addressed by existing work!

Challenge: the "label-switching" problem

Equivalent re-labelings impede mixing

→ slow meeting

Labeling 1



Labeling 2

Key idea: Develop a coupling that is agnostic to the labeling

Coupling Gibbs Over Partitions via Optimal Transport

Coupling Gibbs Over Partitions via Optimal Transport

We frame Gibbs samplers as over partitions instead of over labelings

Tosh and Dasgupta [2014]

Coupling Gibbs Over Partitions via Optimal Transport

We frame Gibbs samplers as over <u>partitions</u> instead of over labelings

- $X \sim p_\Pi(\cdot)$ is a random partition (e.g. $X = \{\{1,3\},\{2\}\}$)
- Define $X(-n)$ as leaving out n (e.g. $X(-1) = \{\{2\},\{3\}\}$)

Tosh and Dasgupta [2014]

Coupling Gibbs Over Partitions via Optimal Transport

We frame Gibbs samplers as over <u>partitions</u> instead of over labelings

- $X \sim p_\Pi(\cdot)$ is a random partition (e.g. $X = \{\{1,3\},\{2\}\}$)
- Define $X(-n)$ as leaving out n (e.g. $X(-1) = \{\{2\},\{3\}\}$)
- Gibbs transition kernel: $X_{t+1} \sim p_{\Pi|\Pi(-n)}\big(\cdot\,|X_t(-n)\big)$

Tosh and Dasgupta [2014]

Coupling Gibbs Over Partitions via Optimal Transport

We frame Gibbs samplers as over partitions instead of over labelings

- $X \sim p_\Pi(\cdot)$ is a random partition (e.g. $X = \{\{1,3\},\{2\}\}$)
- Define $X(-n)$ as leaving out n (e.g. $X(-1) = \{\{2\},\{3\}\}$)
- Gibbs transition kernel: $X_{t+1} \sim p_{\Pi|\Pi(-n)}\big(\cdot\,|X_t(-n)\big) \quad = \sum_k a_k \delta_{\pi^k}(\cdot)$

Tosh and Dasgupta [2014]

Coupling Gibbs Over Partitions via Optimal Transport

We frame Gibbs samplers as over <u>partitions</u> instead of over labelings

- $X \sim p_\Pi(\cdot)$ is a random partition (e.g. $X = \{\{1,3\},\{2\}\}$)
- Define $X(-n)$ as leaving out n (e.g. $X(-1) = \{\{2\},\{3\}\}$)
- Gibbs transition kernel: $X_{t+1} \sim p_{\Pi|\Pi(-n)}\big(\cdot\,|X_t(-n)\big) \quad = \sum_k a_k \delta_{\pi^k}(\cdot)$

$$Y_t \sim p_{\Pi|\Pi(-n)}\big(\cdot\,|Y_{t-1}(-n)\big) = \sum_{k'} b_{k'} \delta_{\nu^{k'}}(\cdot)$$

Tosh and Dasgupta [2014]

Coupling Gibbs Over Partitions via Optimal Transport

We frame Gibbs samplers as over <u>partitions</u> instead of over labelings

- $X \sim p_\Pi(\cdot)$ is a random partition (e.g. $X = \{\{1,3\},\{2\}\}$)
- Define $X(-n)$ as leaving out n (e.g. $X(-1) = \{\{2\},\{3\}\}$)
- Gibbs transition kernel: $X_{t+1} \sim p_{\Pi|\Pi(-n)}\big(\cdot\,|X_t(-n)\big) \quad = \sum_k a_k \delta_{\pi^k}(\cdot)$

$$Y_t \sim p_{\Pi|\Pi(-n)}\big(\cdot\,|Y_{t-1}(-n)\big) = \sum_{k'} b_{k'} \delta_{\nu^{k'}}(\cdot)$$

To couple X_{t+1} and Y_t, we use optimal transport:

$$(X_{t+1}, Y_t) \sim \boldsymbol{\gamma}^*\big(\, p(\,\cdot\,|X_t),\ p(\cdot\,|Y_{t-1})\big)$$

Tosh and Dasgupta [2014]

Coupling Gibbs Over Partitions via Optimal Transport

We frame Gibbs samplers as over <u>partitions</u> instead of over labelings

- $X \sim p_\Pi(\cdot)$ is a random partition (e.g. $X = \{\{1,3\},\{2\}\}$)
- Define $X(-n)$ as leaving out n (e.g. $X(-1) = \{\{2\},\{3\}\}$)
- Gibbs transition kernel: $X_{t+1} \sim p_{\Pi|\Pi(-n)}\big(\cdot\,|X_t(-n)\big) \quad = \sum_k a_k \delta_{\pi^k}(\cdot)$

$$Y_t \sim p_{\Pi|\Pi(-n)}\big(\cdot\,|Y_{t-1}(-n)\big) = \sum_{k'} b_{k'} \delta_{\nu^{k'}}(\cdot)$$

To couple X_{t+1} and Y_t, we use optimal transport:

$$(X_{t+1}, Y_t) \sim \boldsymbol{\gamma}^*(\,p(\,\cdot\,|X_t),\ p(\cdot\,|Y_{t-1}))$$

Strategy for $\boldsymbol{\gamma}^*$: make X_{t+1} and Y_t as <u>close</u> as possible

Tosh and Dasgupta [2014]

Coupling Gibbs Over Partitions via Optimal Transport

We frame Gibbs samplers as over partitions instead of over labelings

- $X \sim p_\Pi(\cdot)$ is a random partition (e.g. $X = \{\{1,3\},\{2\}\}$)
- Define $X(-n)$ as leaving out n (e.g. $X(-1) = \{\{2\},\{3\}\}$)
- Gibbs transition kernel: $X_{t+1} \sim p_{\Pi|\Pi(-n)}\big(\cdot\,|X_t(-n)\big) \quad = \sum_k a_k \delta_{\pi^k}(\cdot)$

$$Y_t \sim p_{\Pi|\Pi(-n)}\big(\cdot\,|Y_{t-1}(-n)\big) = \sum_{k'} b_{k'} \delta_{\nu^{k'}}(\cdot)$$

To couple X_{t+1} and Y_t, we use optimal transport:

$$(X_{t+1}, Y_t) \sim \boldsymbol{\gamma}^*(\,p(\,\cdot\,|X_t),\ p(\cdot\,|Y_{t-1}))$$

Strategy for $\boldsymbol{\gamma}^*$: make X_{t+1} and Y_t as close as possible

Need a metric:

Tosh and Dasgupta [2014]

Coupling Gibbs Over Partitions via Optimal Transport

We frame Gibbs samplers as over partitions instead of over labelings

- $X \sim p_\Pi(\cdot)$ is a random partition (e.g. $X = \{\{1,3\},\{2\}\}$)
- Define $X(-n)$ as leaving out n (e.g. $X(-1) = \{\{2\},\{3\}\}$)
- Gibbs transition kernel: $X_{t+1} \sim p_{\Pi|\Pi(-n)}\big(\cdot\,|X_t(-n)\big) \quad = \sum_k a_k \delta_{\pi^k}(\cdot)$

$$Y_t \sim p_{\Pi|\Pi(-n)}\big(\cdot\,|Y_{t-1}(-n)\big) = \sum_{k'} b_{k'} \delta_{\nu^{k'}}(\cdot)$$

To couple X_{t+1} and Y_t, we use optimal transport:

$$(X_{t+1}, Y_t) \sim \boldsymbol{\gamma}^*(\,p(\,\cdot\,|X_t),\; p(\cdot\,|Y_{t-1}))$$

Strategy for $\boldsymbol{\gamma}^*$: make X_{t+1} and Y_t as close as possible

Need a metric: Use adjacency matrix → Hamming distance

Tosh and Dasgupta [2014]; Rand [1971]

Coupling Gibbs Over Partitions via Optimal Transport

We frame Gibbs samplers as over partitions instead of over labelings

- $X \sim p_\Pi(\cdot)$ is a random partition (e.g. $X = \{\{1,3\},\{2\}\}$)
- Define $X(-n)$ as leaving out n (e.g. $X(-1) = \{\{2\},\{3\}\}$)
- Gibbs transition kernel: $X_{t+1} \sim p_{\Pi|\Pi(-n)}\big(\cdot\,|X_t(-n)\big) = \sum_k a_k \delta_{\pi^k}(\cdot)$

$$Y_t \sim p_{\Pi|\Pi(-n)}\big(\cdot\,|Y_{t-1}(-n)\big) = \sum_{k'} b_{k'} \delta_{\nu^{k'}}(\cdot)$$

To couple X_{t+1} and Y_t, we use optimal transport:

$$(X_{t+1}, Y_t) \sim \boldsymbol{\gamma}^*(\,p(\,\cdot\,|X_t),\ p(\cdot\,|Y_{t-1}))$$

Strategy for $\boldsymbol{\gamma}^*$: make X_{t+1} and Y_t as close as possible

Need a metric: Use adjacency matrix → Hamming distance

OT Problem:

$$\gamma^* = \inf_\gamma \sum_k \sum_{k'} \gamma\big(\pi^k, \nu^{k'}\big) d_{\text{Hamming}}\,(\pi^k, \nu^{k'})$$

$$\text{s.t. } \gamma \geq 0, \sum_k \gamma\big(\pi^k, \nu^{k'}\big) = b^{k'}, \sum_{k'} \gamma\big(\pi^k, \nu^{k'}\big) = a^k$$

Tosh and Dasgupta [2014]; Rand [1971]; Nguyen, **Trippe**, Broderick [2022]

Coupling Gibbs Over Partitions via Optimal Transport

We frame Gibbs samplers as over partitions instead of over labelings

- $X \sim p_\Pi(\cdot)$ is a random partition (e.g. $X = \{\{1,3\},\{2\}\}$)
- Define $X(-n)$ as leaving out n (e.g. $X(-1) = \{\{2\},\{3\}\}$)
- Gibbs transition kernel: $X_{t+1} \sim p_{\Pi|\Pi(-n)}\left(\cdot\,|X_t(-n)\right) = \sum_k a_k \delta_{\pi^k}(\cdot)$

$$Y_t \sim p_{\Pi|\Pi(-n)}\left(\cdot\,|Y_{t-1}(-n)\right) = \sum_{k'} b_{k'} \delta_{\nu^{k'}}(\cdot)$$

To couple X_{t+1} and Y_t, we use optimal transport:

$$(X_{t+1}, Y_t) \sim \boldsymbol{\gamma}^*(\,p(\,\cdot\,|X_t),\ p(\cdot\,|Y_{t-1}))$$

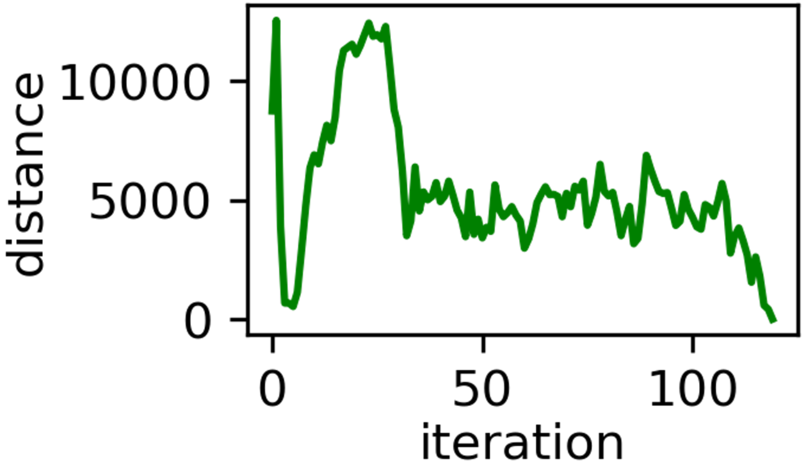
Strategy for $\boldsymbol{\gamma}^*$: make X_{t+1} and Y_t as close as possible

Need a metric: Use adjacency matrix → Hamming distance

OT Problem:

$$\gamma^* = \inf_\gamma \sum_k \sum_{k'} \gamma(\pi^k, \nu^{k'}) d_{\text{Hamming}}(\pi^k, \nu^{k'})$$

$$\text{s.t. } \gamma \geq 0, \sum_k \gamma\left(\pi^k, \nu^{k'}\right) = b^{k'}, \sum_{k'} \gamma\left(\pi^k, \nu^{k'}\right) = a^k$$



Tosh and Dasgupta [2014]; Rand [1971]; Nguyen, **Trippe**, Broderick [2022]

Coupling Gibbs Over Partitions via Optimal Transport

We frame Gibbs samplers as over partitions instead of over labelings

- $X \sim p_\Pi(\cdot)$ is a random partition (e.g. $X = \{\{1,3\},\{2\}\}$)
- Define $X(-n)$ as leaving out n (e.g. $X(-1) = \{\{2\},\{3\}\}$)
- Gibbs transition kernel: $X_{t+1} \sim p_{\Pi|\Pi(-n)}(\cdot\,|X_t(-n)) = \sum_k a_k \delta_{\pi^k}(\cdot)$

$$Y_t \sim p_{\Pi|\Pi(-n)}(\cdot\,|Y_{t-1}(-n)) = \sum_{k'} b_{k'} \delta_{\nu^{k'}}(\cdot)$$

To couple X_{t+1} and Y_t, we use optimal transport:

$$(X_{t+1}, Y_t) \sim \boldsymbol{\gamma}^*(\,p(\cdot\,|X_t),\ p(\cdot\,|Y_{t-1}))$$

Strategy for $\boldsymbol{\gamma}^*$: make X_{t+1} and Y_t as close as possible

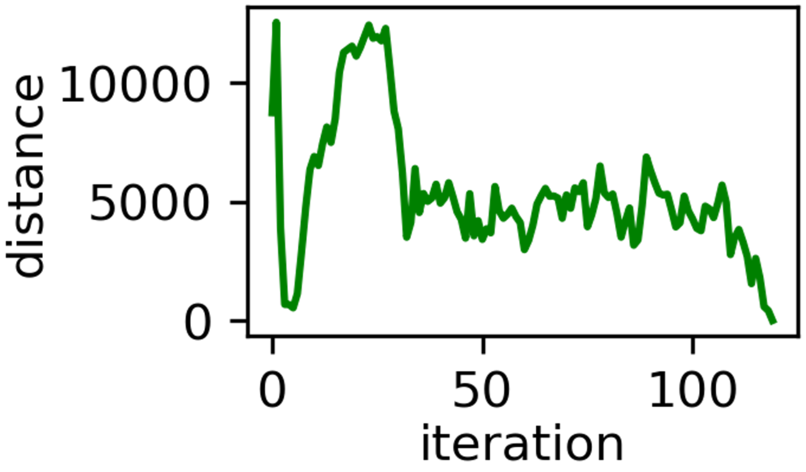
Need a metric: Use adjacency matrix → Hamming distance

OT Problem:

$$\gamma^* = \inf_\gamma \sum_k \sum_{k'} \gamma(\pi^k, \nu^{k'}) d_{\text{Hamming}}(\pi^k, \nu^{k'})$$

$$\text{s.t. } \gamma \geq 0, \sum_k \gamma(\pi^k, \nu^{k'}) = b^{k'}, \sum_{k'} \gamma(\pi^k, \nu^{k'}) = a^k$$

- By construction, does not suffer from label switching!



Tosh and Dasgupta [2014]; Rand [1971]; Nguyen, **Trippe**, Broderick [2022]

Coupling Gibbs Over Partitions via Optimal Transport

We frame Gibbs samplers as over partitions instead of over labelings

- $X \sim p_\Pi(\cdot)$ is a random partition (e.g. $X = \{\{1,3\},\{2\}\}$)
- Define $X(-n)$ as leaving out n (e.g. $X(-1) = \{\{2\},\{3\}\}$)
- Gibbs transition kernel: $X_{t+1} \sim p_{\Pi|\Pi(-n)}(\cdot\,|X_t(-n)) = \sum_k a_k \delta_{\pi^k}(\cdot)$

$$Y_t \sim p_{\Pi|\Pi(-n)}(\cdot\,|Y_{t-1}(-n)) = \sum_{k'} b_{k'} \delta_{\nu^{k'}}(\cdot)$$

To couple X_{t+1} and Y_t, we use optimal transport:

$$(X_{t+1}, Y_t) \sim \boldsymbol{\gamma}^*(\,p(\cdot\,|X_t),\ p(\cdot\,|Y_{t-1}))$$

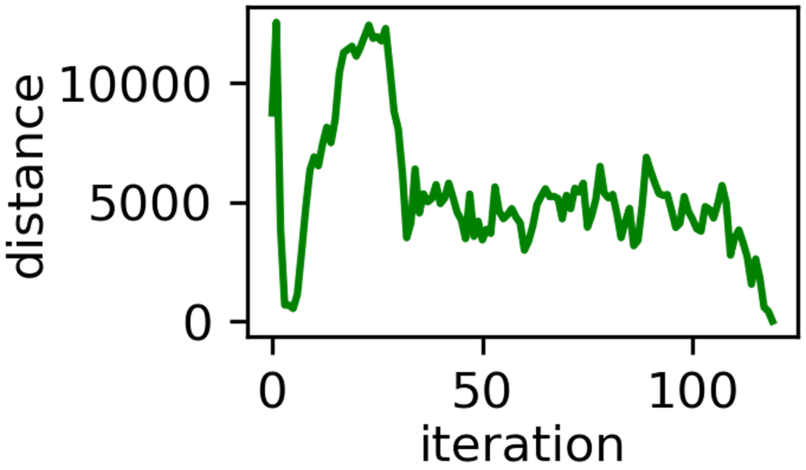
Strategy for $\boldsymbol{\gamma}^*$: make X_{t+1} and Y_t as close as possible

Need a metric: Use adjacency matrix → Hamming distance

OT Problem:

$$\gamma^* = \inf_\gamma \sum_k \sum_{k'} \gamma(\pi^k, \nu^{k'}) d_{\text{Hamming}}(\pi^k, \nu^{k'})$$

$$\text{s.t. } \gamma \geq 0, \sum_k \gamma(\pi^k, \nu^{k'}) = b^{k'}, \sum_{k'} \gamma(\pi^k, \nu^{k'}) = a^k$$



- By construction, does not suffer from label switching!
- **We prove:** $\boldsymbol{\gamma}^*$ permits unbiased estimation

Tosh and Dasgupta [2014]; Rand [1971]; Nguyen, **Trippe**, Broderick [2022]

Our OT coupling meets quickly by avoiding label-switching

Single-cell clustering – Dirichlet process mixture model

- Run many pairs of coupled chains → compute meeting time distribution

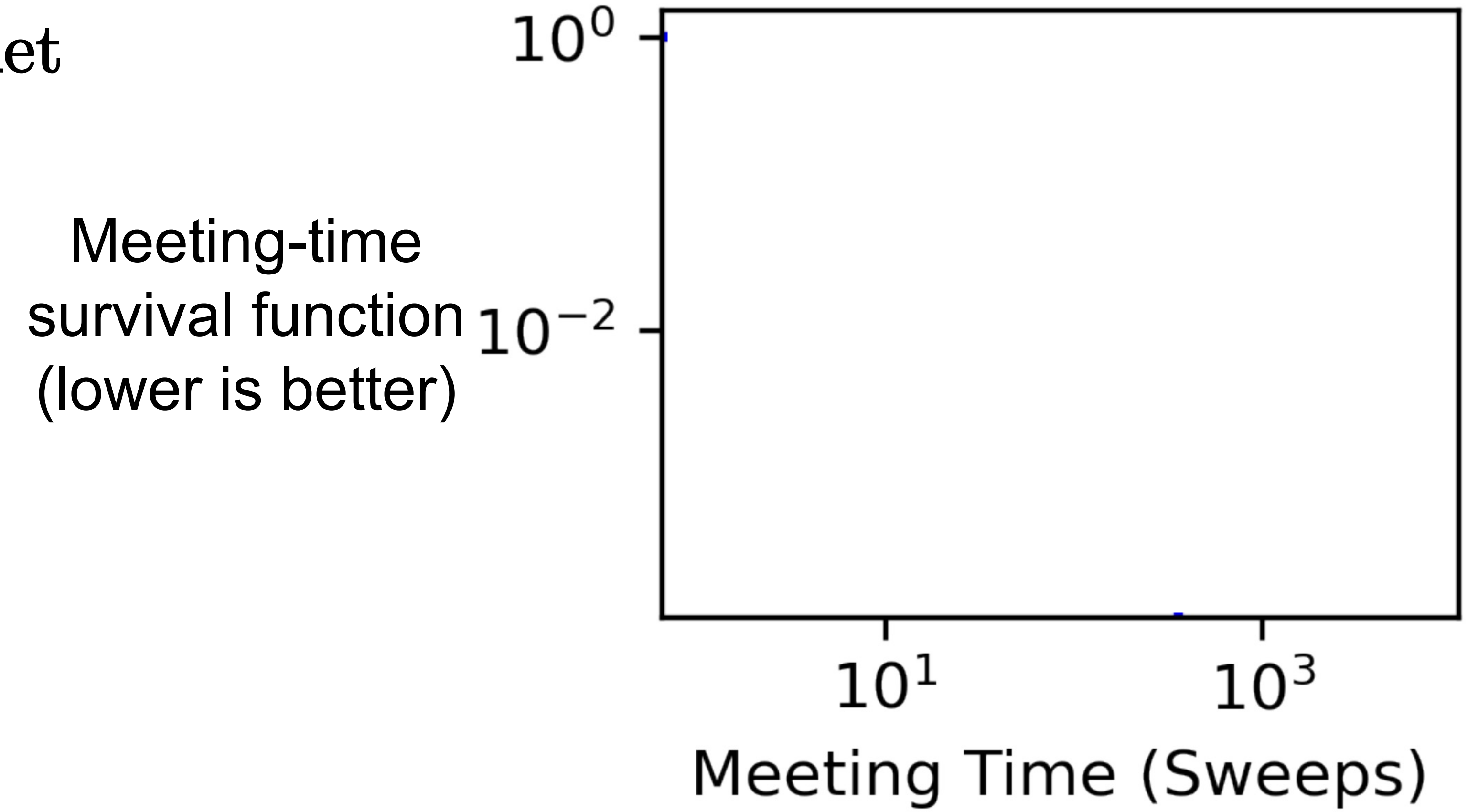
Our OT coupling meets quickly by avoiding label-switching

Single-cell clustering – Dirichlet process mixture model

- Run many pairs of coupled chains → compute meeting time distribution

Coupling meeting-time

OT vs. label-based



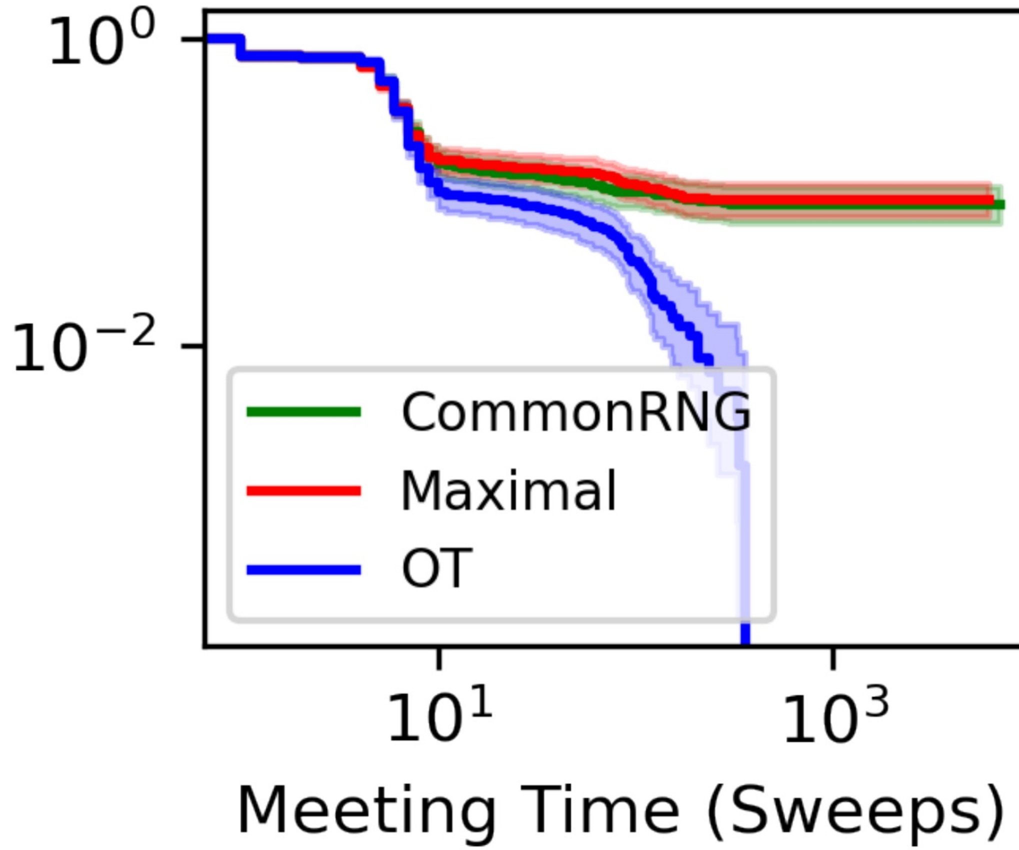
Our OT coupling meets quickly by avoiding label-switching

Single-cell clustering – Dirichlet process mixture model

- Run many pairs of coupled chains → compute meeting time distribution

Coupling meeting-time
OT vs. label-based

Meeting-time survival function (lower is better)



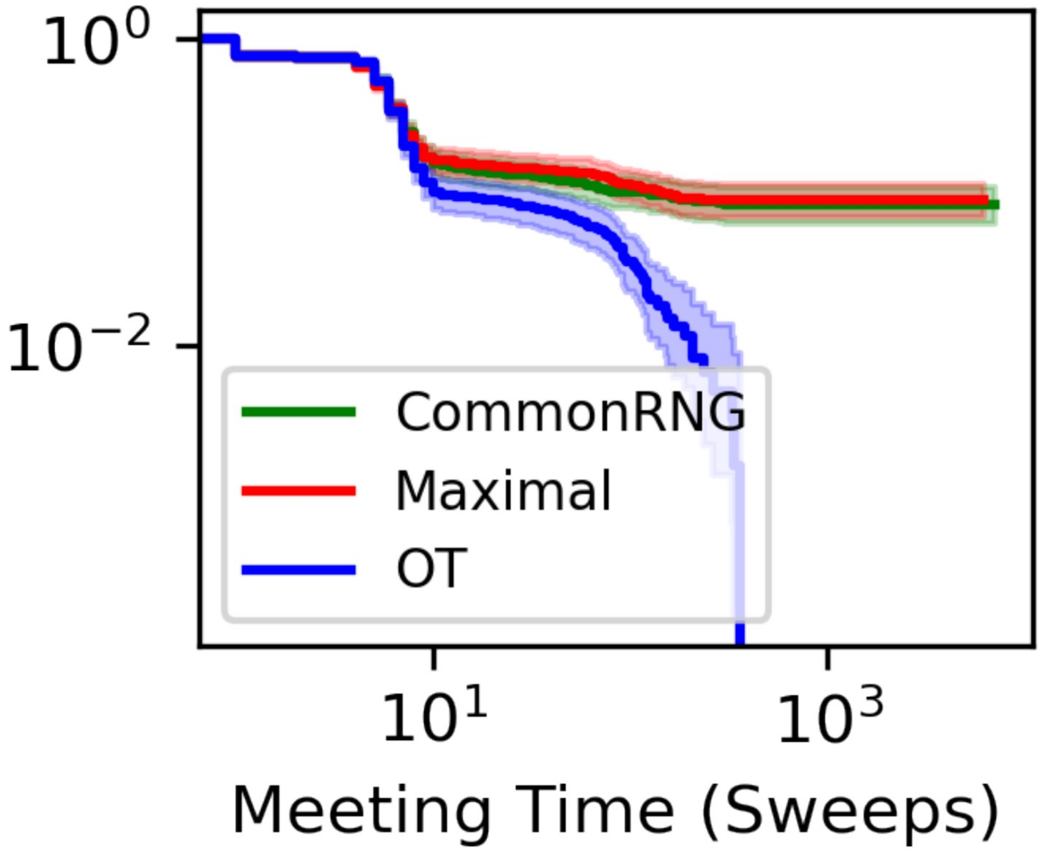
Our OT coupling meets quickly by avoiding label-switching

Single-cell clustering – Dirichlet process mixture model

- Run many pairs of coupled chains → compute meeting time distribution
- Consider label-based couplings:
  - Common RNG (Gibbs, 2004)
  - Maximal coupling (Jerrum, 1998)

Meeting-time survival function (lower is better)

Coupling meeting-time
OT vs. label-based



Roadmap

- Parallelizing MCMC with Couplings:
  - Background & Notation
  - The Label Switching Problem
- We Frame Gibbs Sampling as Markov Chain on Partitions
- Our Optimal Transport Coupling
- **Big-O Analysis Demonstrates Fast Computation**
- Improved Estimation Error and Intervals with OTC over Naïve Parallelism in Practice

Can OT-couplings compete with Naïve use of parallelism?

Can OT-couplings compete with Naïve use of parallelism?

Can we compute our coupling fast enough?

- If coupling is too time intensive, we might prefer single chains

Can OT-couplings compete with Naïve use of parallelism?

Can we compute our coupling fast enough?

- If coupling is too time intensive, we might prefer single chains
- Naïve computation of $d_{\text{Hamming}}(\cdot,\cdot) \rightarrow O(N^2)$ time (let alone OT problem)

Can OT-couplings compete with Naïve use of parallelism?

Can we compute our coupling fast enough?

- If coupling is too time intensive, we might prefer single chains
- Naïve computation of $d_{\text{Hamming}}(\cdot,\cdot)$ →$O(N^2)$ time (let alone OT problem)

We show: can compute coupling in $O(K^3 \log K)$ amortized time!

- K: # of clusters --- typically fixed or $O(\log N)$

Can OT-couplings compete with Naïve use of parallelism?

Can we compute our coupling fast enough?

- If coupling is too time intensive, we might prefer single chains
- Naïve computation of $d_{\text{Hamming}}(\cdot,\cdot) \rightarrow O(N^2)$ time (let alone OT problem)

We show: can compute coupling in $O(K^3 \log K)$ amortized time!

- K: # of clusters --- typically fixed or $O(\log N)$
- Bottleneck : Orlin's algorithm in OT problem (but fast in practice)

Can OT-couplings compete with Naïve use of parallelism?

Can we compute our coupling fast enough?

- If coupling is too time intensive, we might prefer single chains
- Naïve computation of $d_{\text{Hamming}}(\cdot,\cdot) \rightarrow O(N^2)$ time (let alone OT problem)

We show: can compute coupling in $O(K^3 \log K)$ amortized time!

- K: # of clusters --- typically fixed or $O(\log N)$
- Bottleneck : Orlin's algorithm in OT problem (but fast in practice)
- Compute cost dominated by marginal kernel in practice

Can OT-couplings compete with Naïve use of parallelism?

Can we compute our coupling fast enough?

- If coupling is too time intensive, we might prefer single chains
- Naïve computation of $d_{\text{Hamming}}(\cdot,\cdot)$ → $O(N^2)$ time (let alone OT problem)

We show: can compute coupling in $O(K^3 \log K)$ amortized time!

- K: # of clusters --- typically fixed or $O(\log N)$
- Bottleneck : Orlin's algorithm in OT problem (but fast in practice)
- Compute cost dominated by marginal kernel in practice

Additional challenge: Higher variance than single chains

Can OT-couplings compete with Naïve use of parallelism?

Can we compute our coupling fast enough?

- If coupling is too time intensive, we might prefer single chains
- Naïve computation of $d_{\text{Hamming}}(\cdot,\cdot)$ →$O(N^2)$ time (let alone OT problem)

We show: can compute coupling in $O(K^3 \log K)$ amortized time!

- K: # of clusters --- typically fixed or $O(\log N)$
- Bottleneck : Orlin's algorithm in OT problem (but fast in practice)
- Compute cost dominated by marginal kernel in practice

Additional challenge: Higher variance than single chains

- How many processors are needed?
- Previous works do not compare to naïve use of parallelism

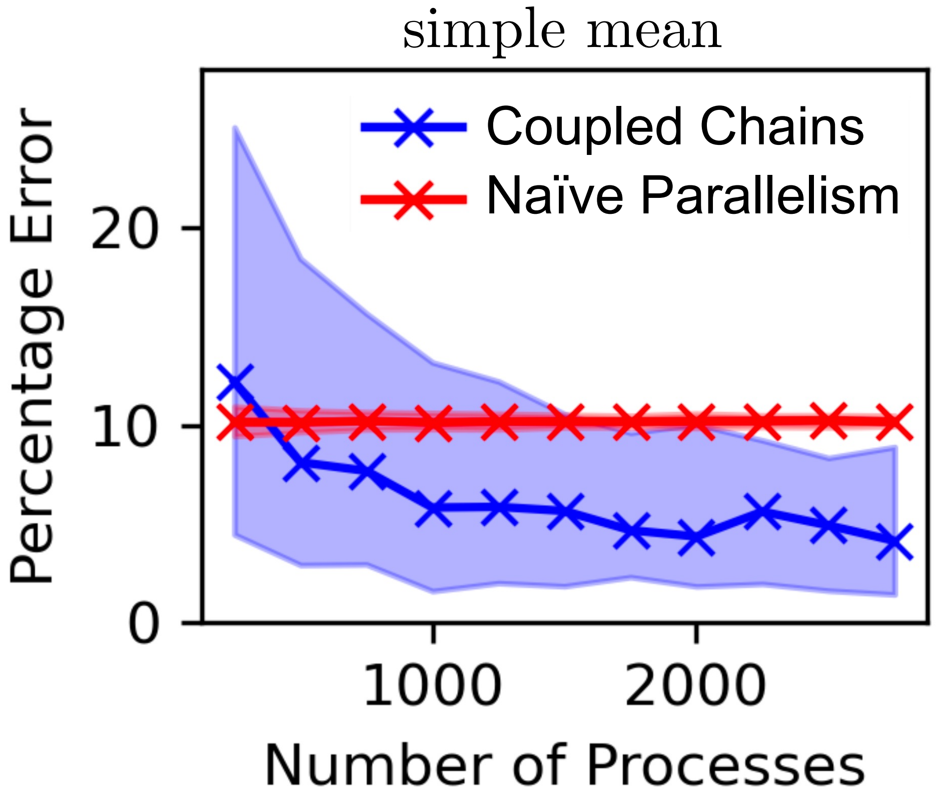
OT-Couplings can be more precise than Naïve parallelism

- Coupled chains: aggregate estimates from multiple pairs of chains
- Naïve parallelism (baseline): average (biased) estimates from single chains

OT-Couplings can be more precise than Naïve parallelism

- Coupled chains: aggregate estimates from multiple pairs of chains
- Naïve parallelism (baseline): average (biased) estimates from single chains

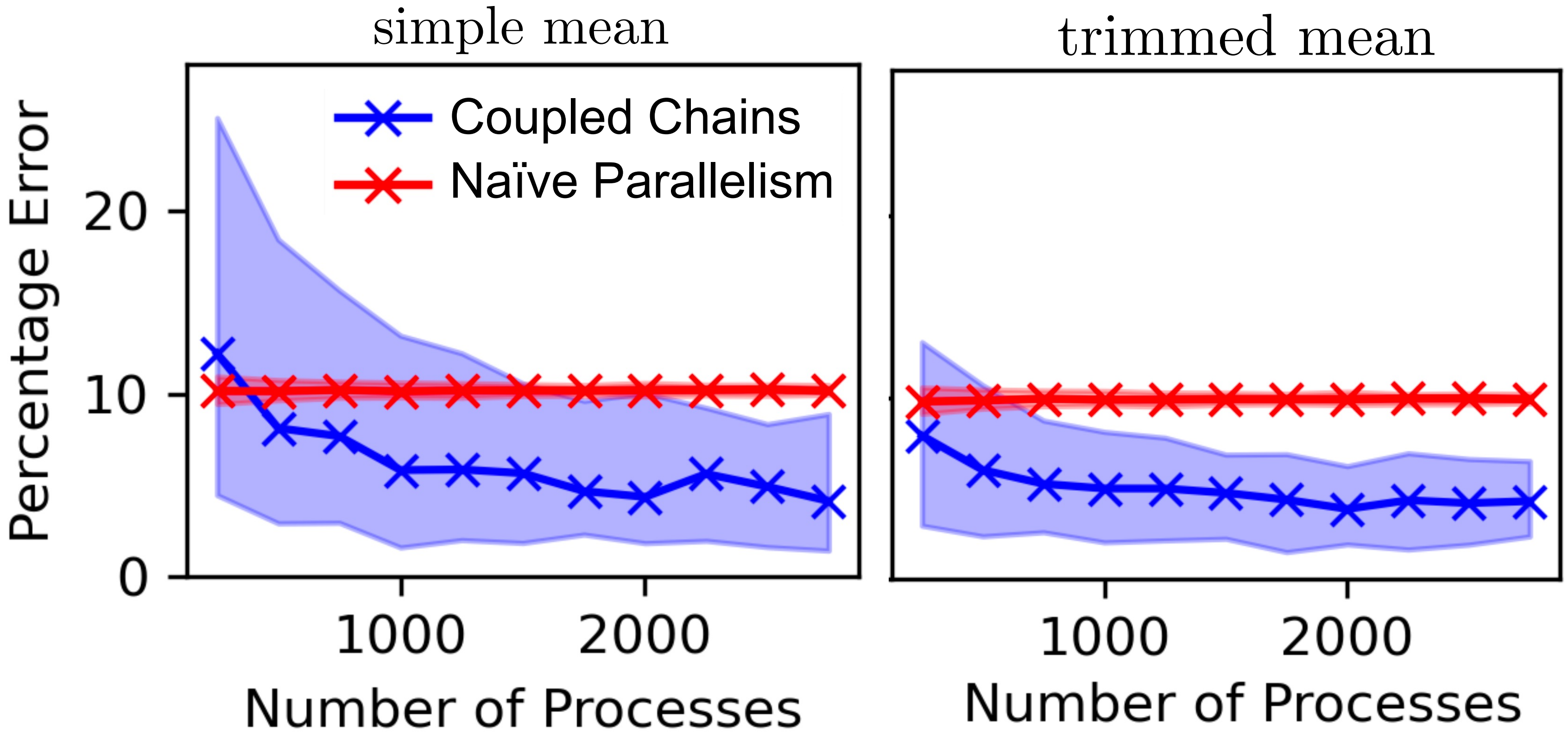
Single-cell clustering (largest component proportion)



OT-Couplings can be more precise than Naïve parallelism

- Coupled chains: aggregate estimates from multiple pairs of chains
- Naïve parallelism (baseline): average (biased) estimates from single chains

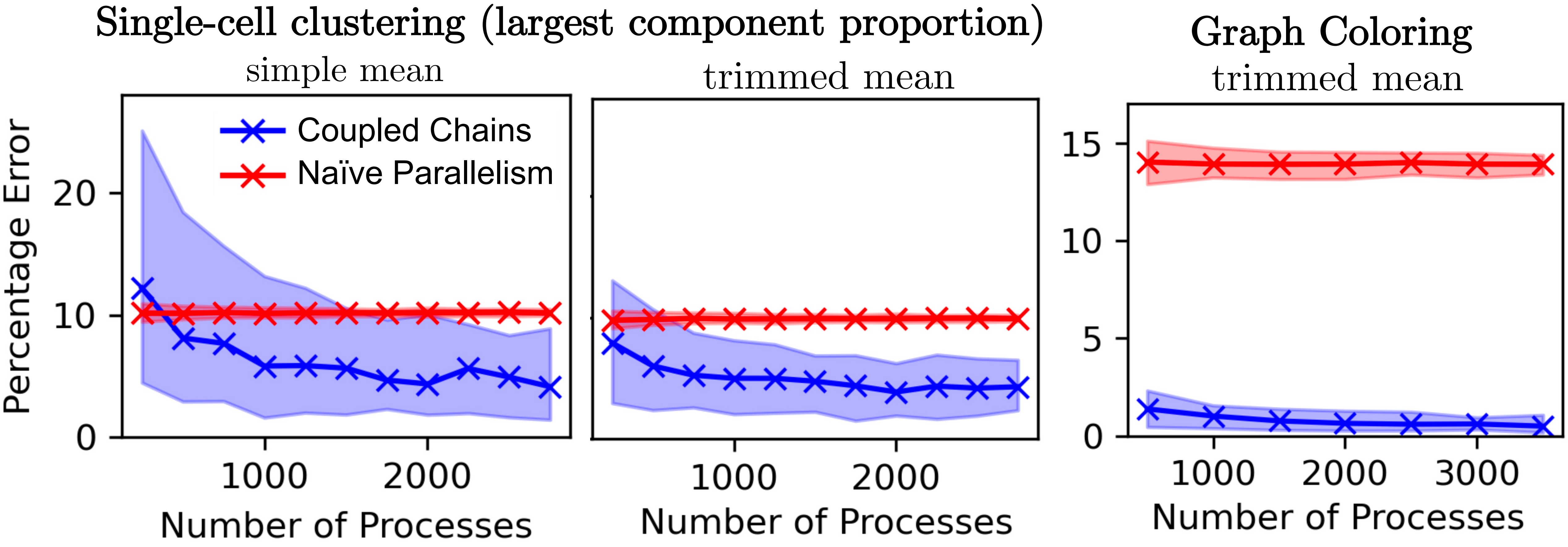
Single-cell clustering (largest component proportion)



- Further improvement with robust estimators (clipping outliers)

OT-Couplings can be more precise than Naïve parallelism

- Coupled chains: aggregate estimates from multiple pairs of chains
- Naïve parallelism (baseline): average (biased) estimates from single chains



- Further improvement with robust estimators (clipping outliers)

OT-Couplings provide reliable confidence intervals

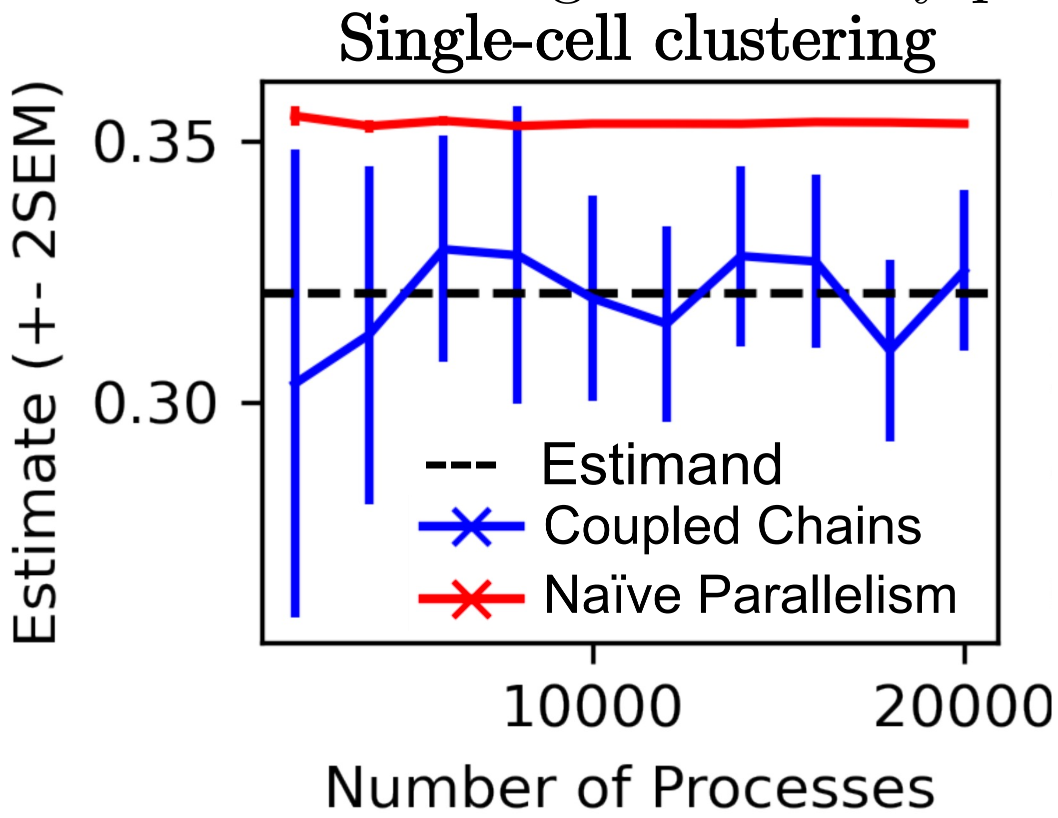
- Each process gives an i.i.d. sample → use standard errors to form confidence intervals

OT-Couplings provide reliable confidence intervals

- Each process gives an i.i.d. sample → use standard errors to form confidence intervals
- Correct coverage with many processors

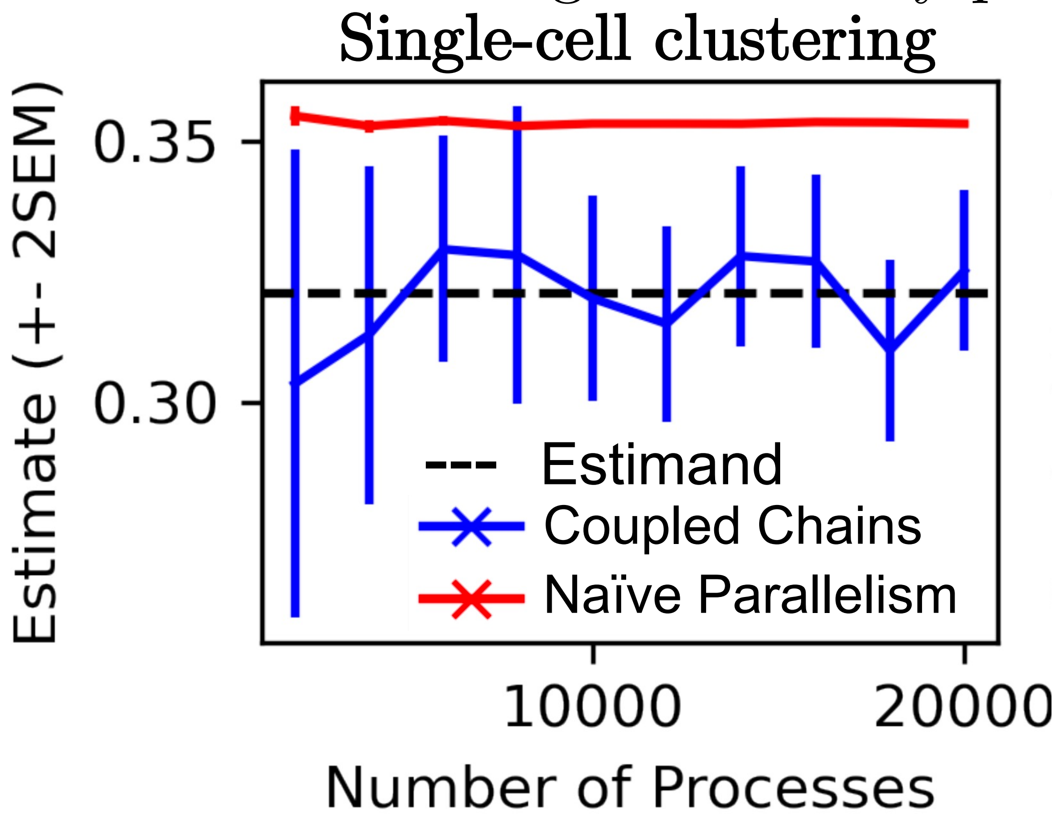
OT-Couplings provide reliable confidence intervals

- Each process gives an i.i.d. sample → use standard errors to form confidence intervals
- Correct coverage with many processors



OT-Couplings provide reliable confidence intervals

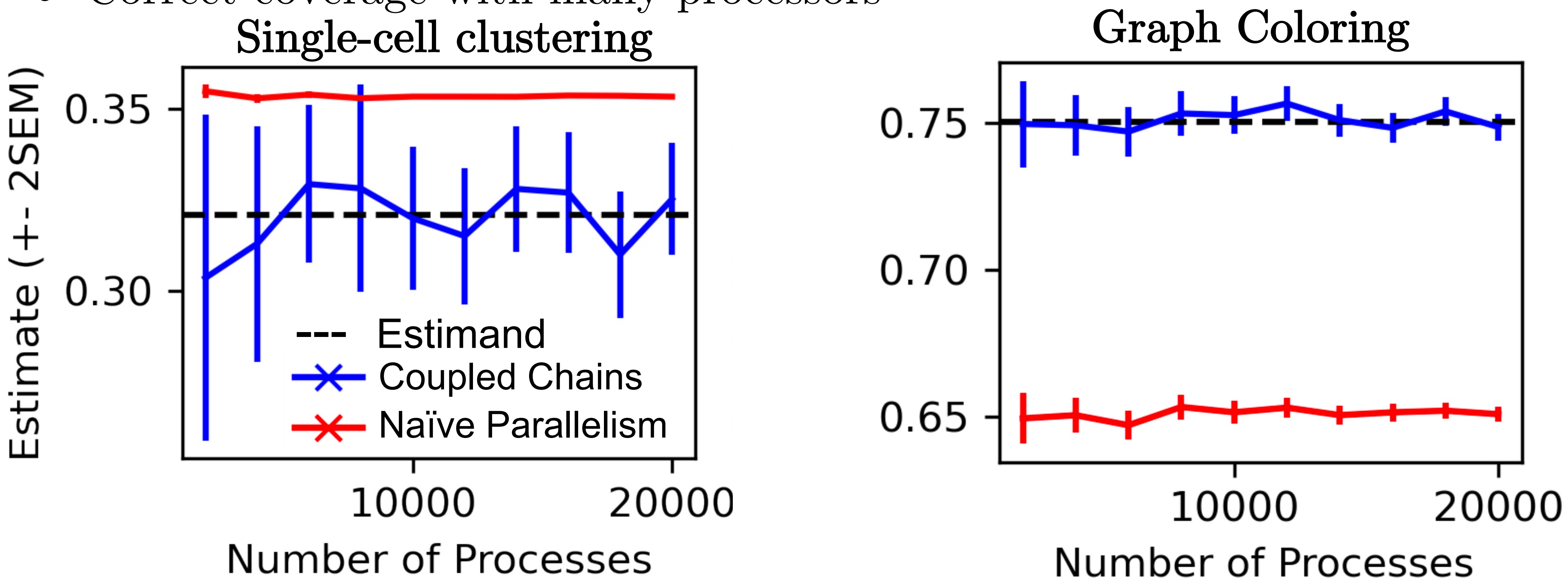
- Each process gives an i.i.d. sample → use standard errors to form confidence intervals
- Correct coverage with many processors



- Analogous "intervals" from single chains do not cover the estimand

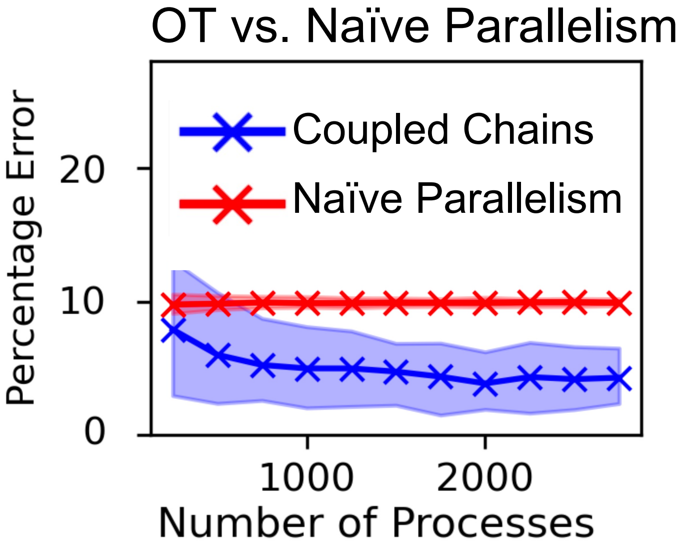
OT-Couplings provide reliable confidence intervals

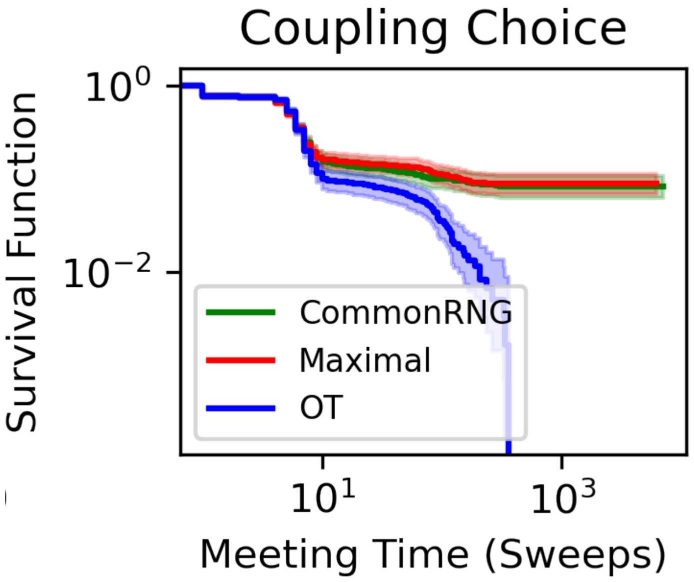
- Each process gives an i.i.d. sample → use standard errors to form confidence intervals
- Correct coverage with many processors

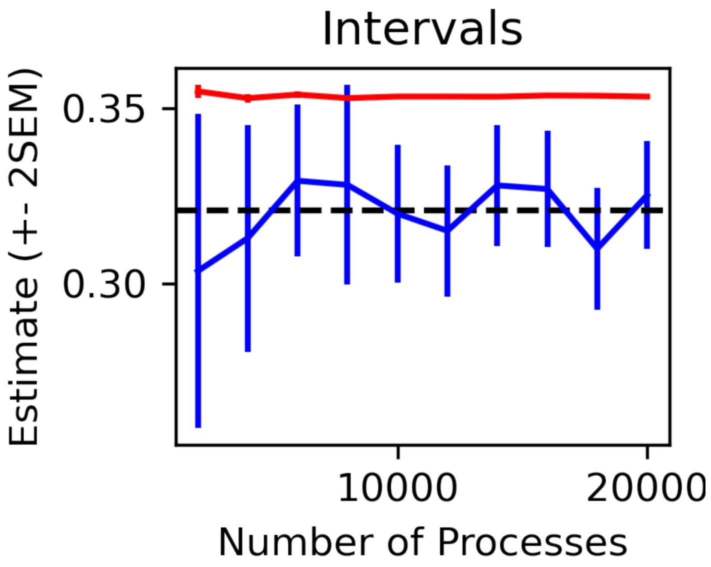


- Analogous "intervals" from single chains do not cover the estimand

Conclusions







- High accuracy via parallelism
- OT avoids label-switching
- Intervals with coverage

Contact: tdn@mit.edu, btrippe@mit.edu, tamarab@mit.edu

Main References:

"Many processors, little time: MCMC for partitions via optimal transport couplings."
Tin Nguyen, **Brian Trippe** & Tamara Broderick in *AISTATS* (2022)

"Optimal transport couplings of Gibbs samplers on partitions for unbiased estimation"
Brian Trippe*, Tin Nguyen* & Tamara Broderick in *AABI* (2021) [*equal contribution]

References Cited: Jacob, O'Leary & Atchadé. "Unbiased Markov chain Monte Carlo methods with couplings." JRSS-B (2020); Tosh & Dasgupta. "Lower bounds for the Gibbs sampler over mixtures of Gaussians." ICML (2014).; Rand "Objective criteria for the evaluation of clustering methods." JASA (1971)