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

Tamara Broderick

Associate Professor,
MIT

With Tin Nguyen, Brian Trippe

Clustering, many processors, little time

Clustering, many processors, little time

- Clustering is widely used: e.g. to discover cell types in genetics or to check for gerrymandering in politics

[Prabhakaran et al 2016, DeFord et al 2021]

Clustering, many processors, little time

- Clustering is widely used: e.g. to discover cell types in genetics or to check for gerrymandering in politics [Prabhakaran et al 2016, DeFord et al 2021]
- In clustering/partitioning, often interested in computing an expectation: e.g. a Bayesian estimate of the proportion of data in the biggest cluster or co-clustering probability

Clustering, many processors, little time

- Clustering is widely used: e.g. to discover cell types in genetics or to check for gerrymandering in politics

[Prabhakaran et al 2016, DeFord et al 2021]

- In clustering/partitioning, often interested in computing an expectation: e.g. a Bayesian estimate of the proportion of data in the biggest cluster or co-clustering probability
- Markov chain Monte Carlo is widely used (e.g. to approximate Bayesian inference), but can be slow

Clustering, many processors, little time

- Clustering is widely used: e.g. to discover cell types in genetics or to check for gerrymandering in politics
[Prabhakaran et al 2016, DeFord et al 2021]
- In clustering/partitioning, often interested in computing an expectation: e.g. a Bayesian estimate of the proportion of data in the biggest cluster or co-clustering probability
- Markov chain Monte Carlo is widely used (e.g. to approximate Bayesian inference), but can be slow
- Naive parallel processing reduces variance but not bias

Clustering, many processors, little time

- Clustering is widely used: e.g. to discover cell types in genetics or to check for gerrymandering in politics

[Prabhakaran et al 2016, DeFord et al 2021]

- In clustering/partitioning, often interested in computing an expectation: e.g. a Bayesian estimate of the proportion of data in the biggest cluster or co-clustering probability
- Markov chain Monte Carlo is widely used (e.g. to approximate Bayesian inference), but can be slow
- Naive parallel processing reduces variance but not bias
- We find naive "coupling" fails, due to "label switching"

[Jacob et al 2020, Xu et al 2021]

Clustering, many processors, little time

- Clustering is widely used: e.g. to discover cell types in genetics or to check for gerrymandering in politics
  [Prabhakaran et al 2016, DeFord et al 2021]
- In clustering/partitioning, often interested in computing an expectation: e.g. a Bayesian estimate of the proportion of data in the biggest cluster or co-clustering probability
- Markov chain Monte Carlo is widely used (e.g. to approximate Bayesian inference), but can be slow
- Naive parallel processing reduces variance but not bias
- We find naive “coupling” fails, due to “label switching”
  [Jacob et al 2020, Xu et al 2021]
- **We develop**: “optimal transport couplings” for partition models to remove bias at a single processor

Clustering, many processors, little time

- Clustering is widely used: e.g. to discover cell types in genetics or to check for gerrymandering in politics [Prabhakaran et al 2016, DeFord et al 2021]
- In clustering/partitioning, often interested in computing an expectation: e.g. a Bayesian estimate of the proportion of data in the biggest cluster or co-clustering probability
- Markov chain Monte Carlo is widely used (e.g. to approximate Bayesian inference), but can be slow
- Naive parallel processing reduces variance but not bias
- We find naive "coupling" fails, due to "label switching" [Jacob et al 2020, Xu et al 2021]
- **We develop**: "optimal transport couplings" for partition models to remove bias at a single processor
- In the time-limited, highly parallel regime, **we show**: substantial accuracy benefits of our method over naive parallelism and naive use of existing coupling ideas

Roadmap

Roadmap

- Setup for learning partitions with many processors

Roadmap

- Setup for learning partitions with many processors
- Couplings for removing bias from MCMC

Roadmap

- Setup for learning partitions with many processors
- Couplings for removing bias from MCMC
- The challenges of couplings with partitions

Roadmap

- Setup for learning partitions with many processors
- Couplings for removing bias from MCMC
- The challenges of couplings with partitions
- Our proposal for couplings with partitions

Roadmap

- Setup for learning partitions with many processors
- Couplings for removing bias from MCMC
- The challenges of couplings with partitions
- Our proposal for couplings with partitions
- Theory that our method is unbiased and fast

Roadmap

- Setup for learning partitions with many processors
- Couplings for removing bias from MCMC
- The challenges of couplings with partitions
- Our proposal for couplings with partitions
- Theory that our method is unbiased and fast
- Experiments: our method gives good estimates and confidence intervals

Roadmap

- Setup for learning partitions with many processors
- Couplings for removing bias from MCMC
- The challenges of couplings with partitions
- Our proposal for coupling with partitions
- Theory that our method is unbiased and fast
- Experiments: our method gives good estimates and confidence intervals

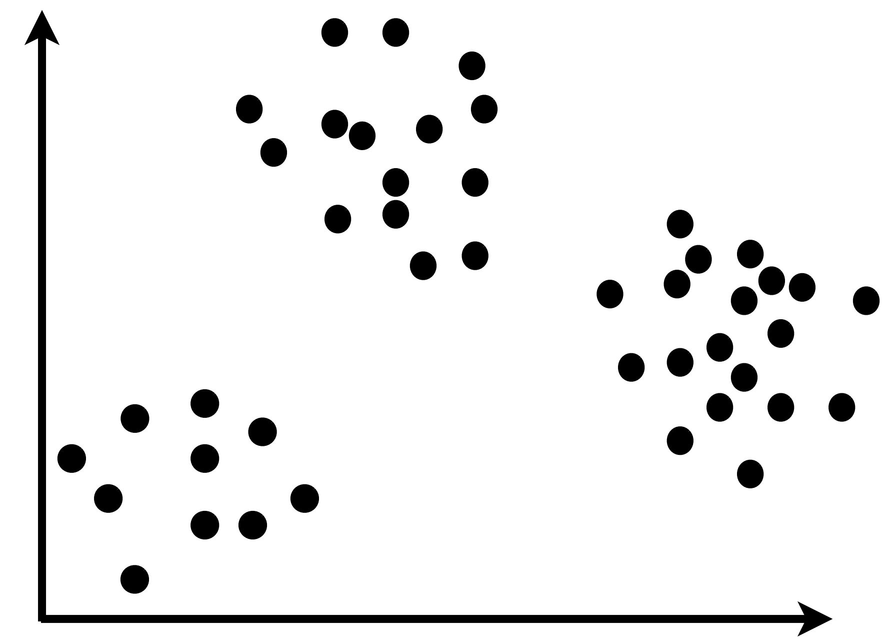
Setup

Setup

- A partition Π assigns the data to mutually exclusive & exhaustive groups

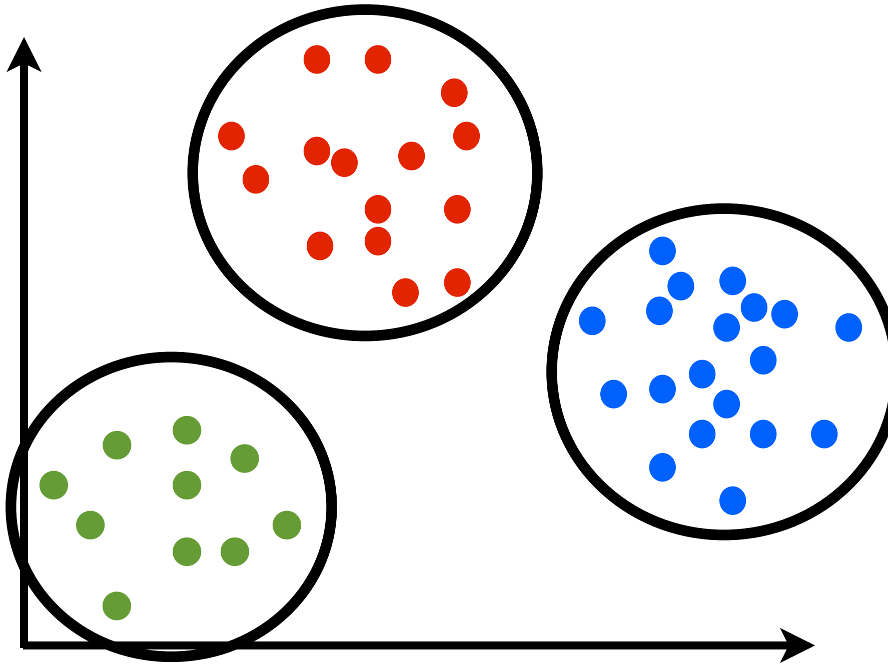
Setup

- A partition Π assigns the data to mutually exclusive & exhaustive groups



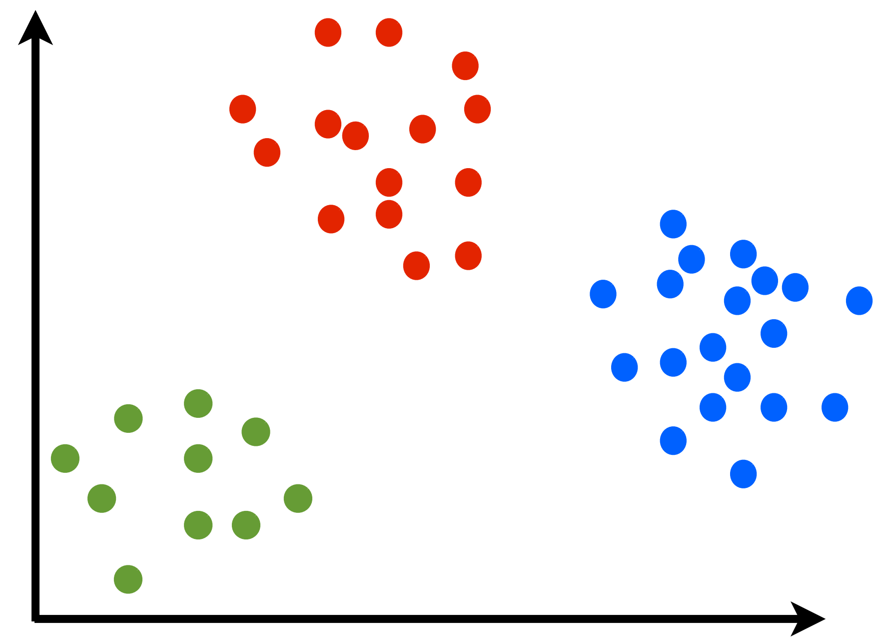
Setup

- A partition Π assigns the data to mutually exclusive & exhaustive groups

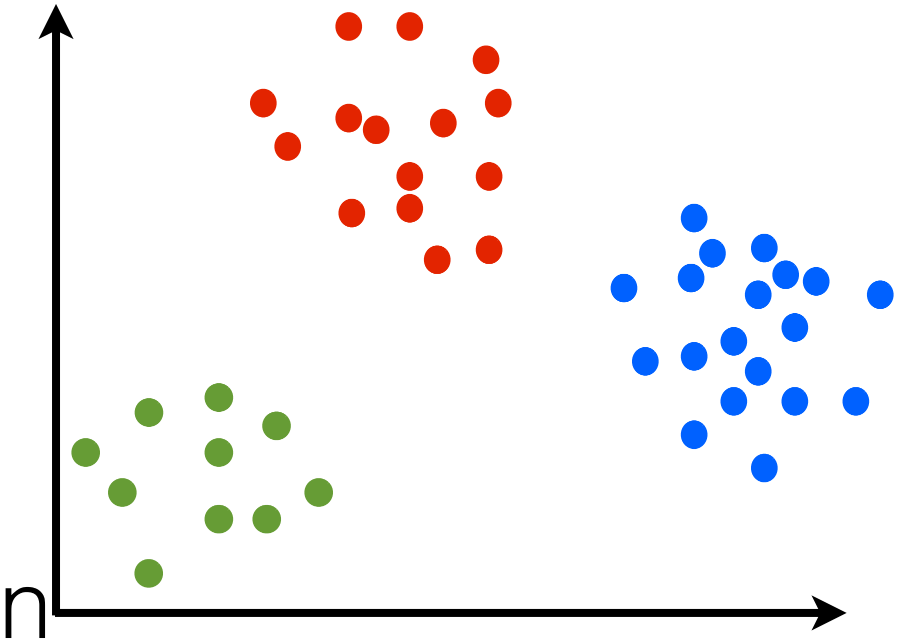


Setup

- A partition Π assigns the data to mutually exclusive & exhaustive groups

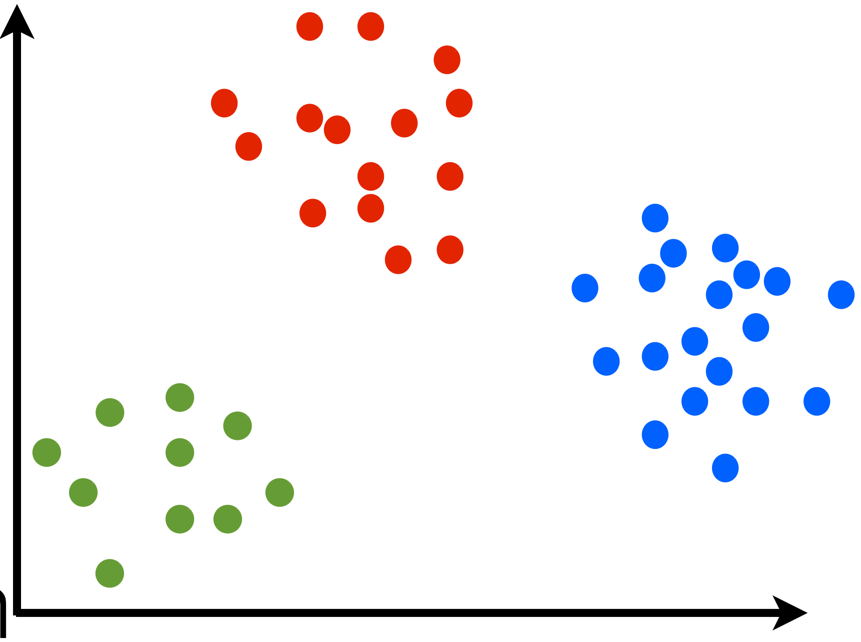


Setup



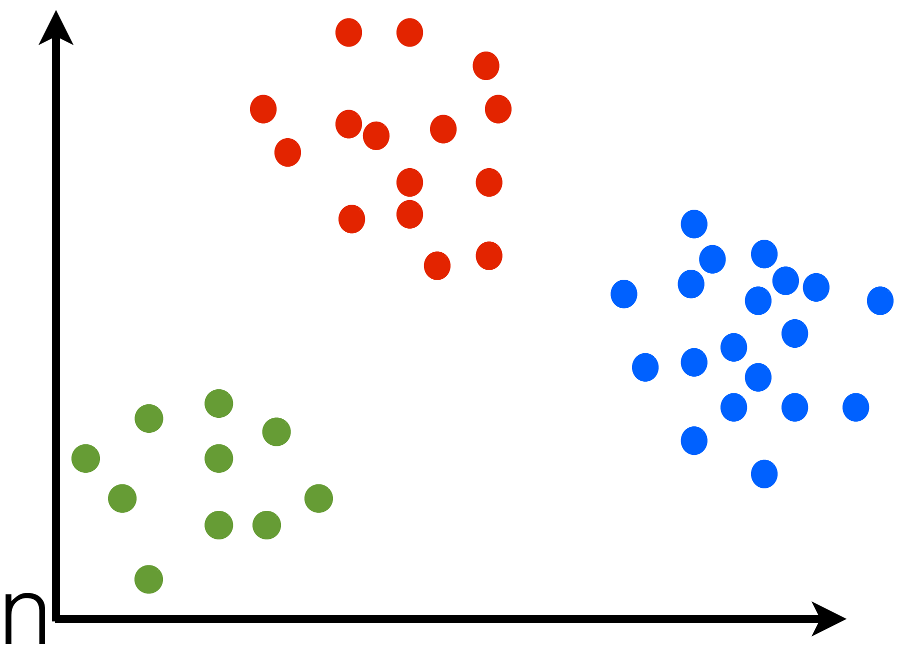
- A partition Π assigns the data to mutually exclusive & exhaustive groups
- Example problem: find a Bayesian estimate of the largest-cluster proportion

Setup



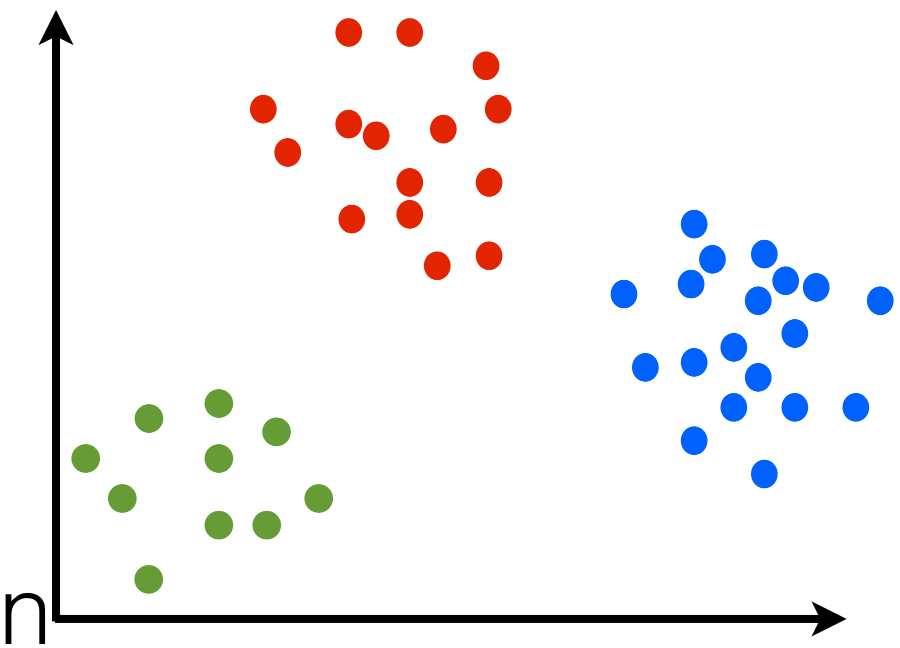
- A partition Π assigns the data to mutually exclusive & exhaustive groups
- Example problem: find a Bayesian estimate of the largest-cluster proportion
  - Let $h(\Pi)$ return proportion of data in the largest cluster

Setup



- A partition Π assigns the data to mutually exclusive & exhaustive groups
- Example problem: find a Bayesian estimate of the largest-cluster proportion
  - Let $h(\Pi)$ return proportion of data in the largest cluster
  - Let p_{Π} be the posterior on Π after observing the data

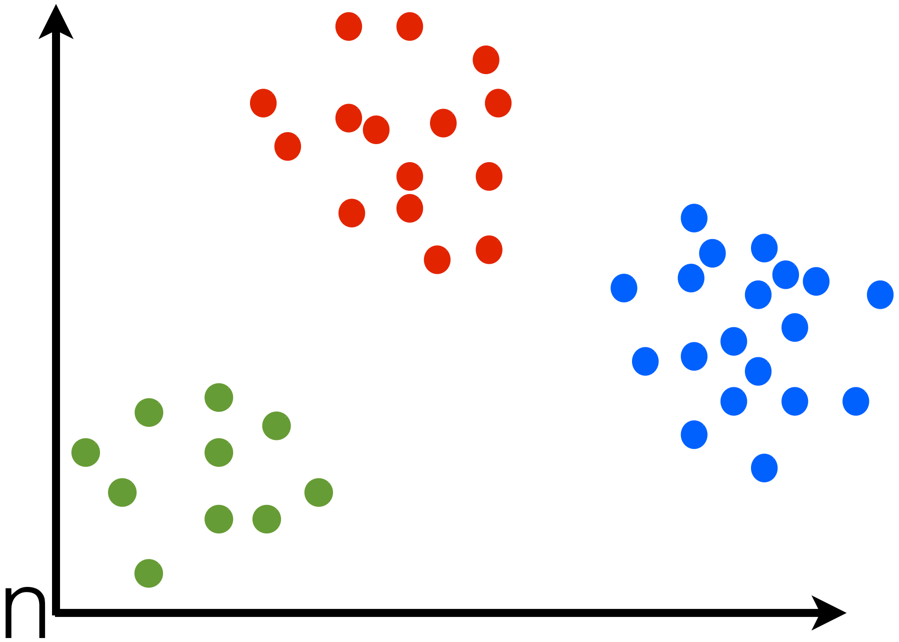
Setup



- A partition Π assigns the data to mutually exclusive & exhaustive groups
- Example problem: find a Bayesian estimate of the largest-cluster proportion
  - Let $h(\Pi)$ return proportion of data in the largest cluster
  - Let p_Π be the posterior on Π after observing the data

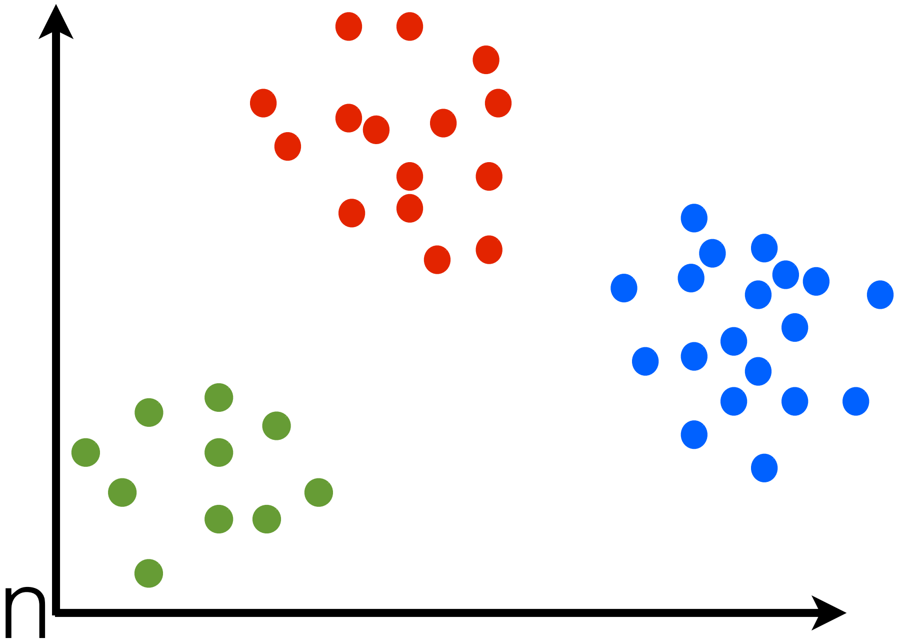
$$H^* = \int h(\Pi) p_\Pi(\Pi) d\Pi$$

Setup



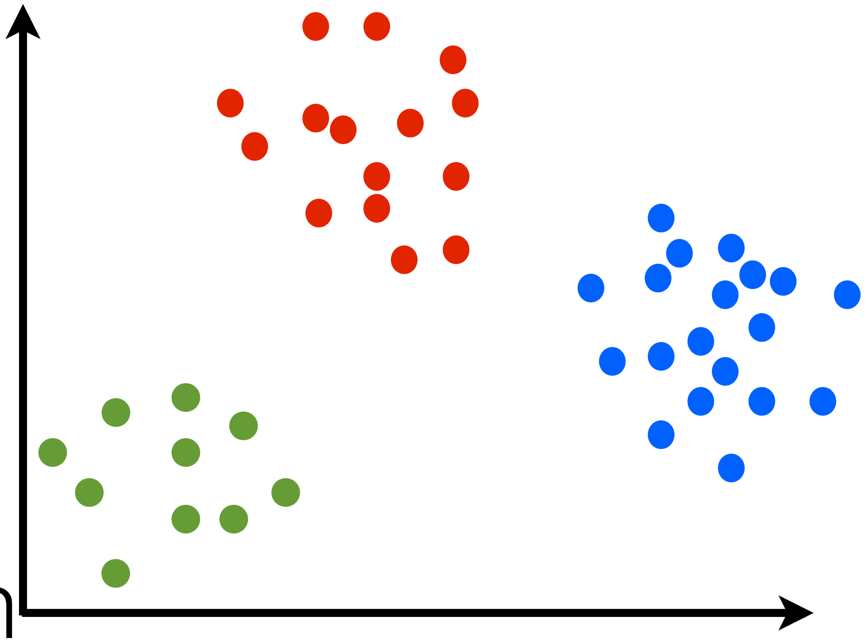
- A partition Π assigns the data to mutually exclusive & exhaustive groups
- Example problem: find a Bayesian estimate of the largest-cluster proportion
  - Let $h(\Pi)$ return proportion of data in the largest cluster
  - Let p_Π be the posterior on Π after observing the data
- Our general goal is to report: $H^* = \int h(\Pi) p_\Pi(\Pi) d\Pi$

Setup



- A partition Π assigns the data to mutually exclusive & exhaustive groups
- Example problem: find a Bayesian estimate of the largest-cluster proportion
    - Let $h(\Pi)$ return proportion of data in the largest cluster
    - Let p_Π be the posterior on Π after observing the data
- Our general goal is to report: $H^* = \int h(\Pi) p_\Pi(\Pi) d\Pi$
- We often need to use a (random) approximation $\hat{H}$ for H^*

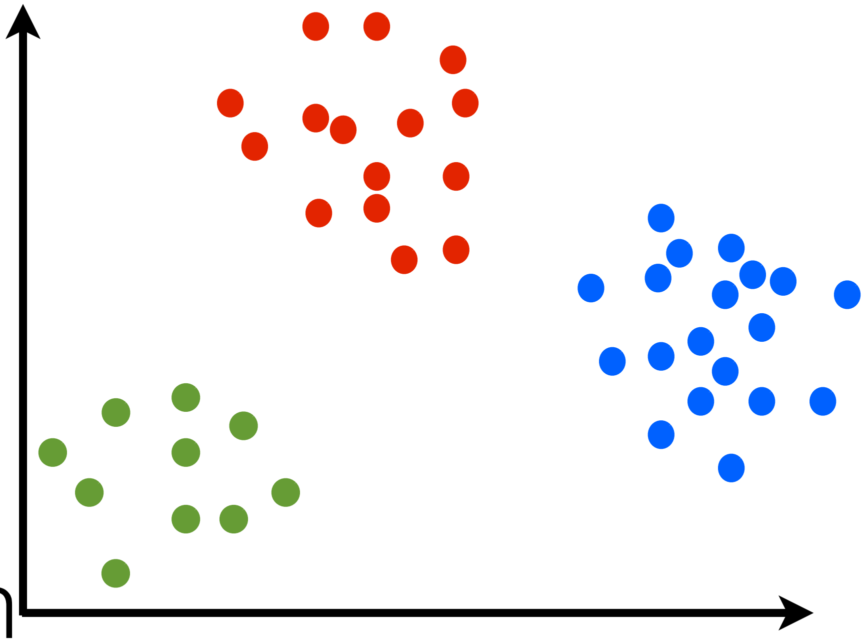
Setup



- A partition Π assigns the data to mutually exclusive & exhaustive groups
- Example problem: find a Bayesian estimate of the largest-cluster proportion
  - Let $h(\Pi)$ return proportion of data in the largest cluster
  - Let p_Π be the posterior on Π after observing the data
- Our general goal is to report: $H^* = \int h(\Pi) p_\Pi(\Pi) d\Pi$
- We often need to use a (random) approximation $\hat{H}$ for H^*
- Mean squared error (of the approx)

$$\mathbb{E}[(\hat{H} - H^*)^2]$$

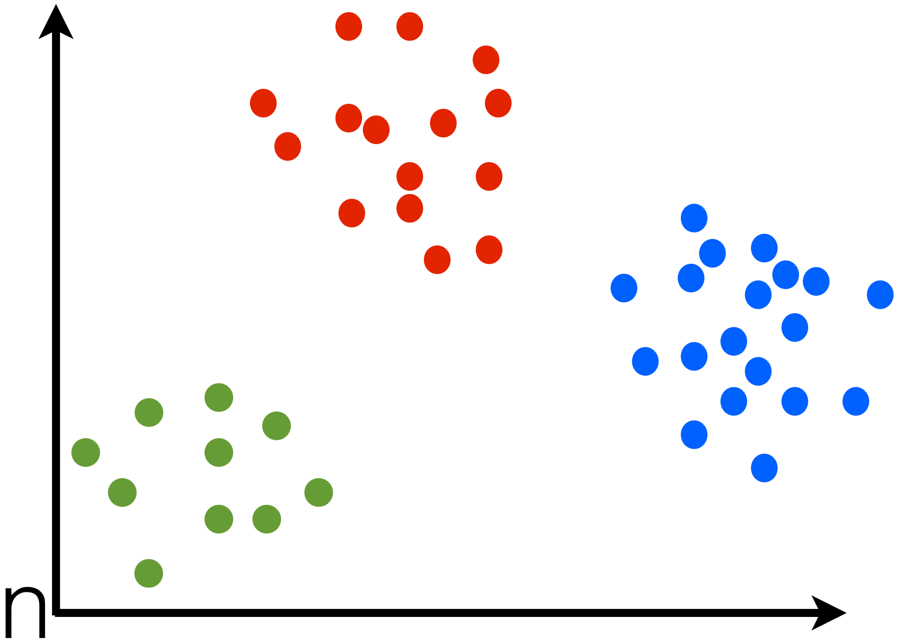
Setup



- A partition Π assigns the data to mutually exclusive & exhaustive groups
- Example problem: find a Bayesian estimate of the largest-cluster proportion
  - Let $h(\Pi)$ return proportion of data in the largest cluster
  - Let p_Π be the posterior on Π after observing the data
- Our general goal is to report: $H^* = \int h(\Pi) p_\Pi(\Pi) d\Pi$
- We often need to use a (random) approximation $\hat{H}$ for H^*
- Mean squared error (of the approx) = bias² + variance

$$\mathbb{E}[(\hat{H} - H^*)^2] = (\mathbb{E}\hat{H} - H^*)^2 + \mathrm{Var}(\hat{H})$$

Setup

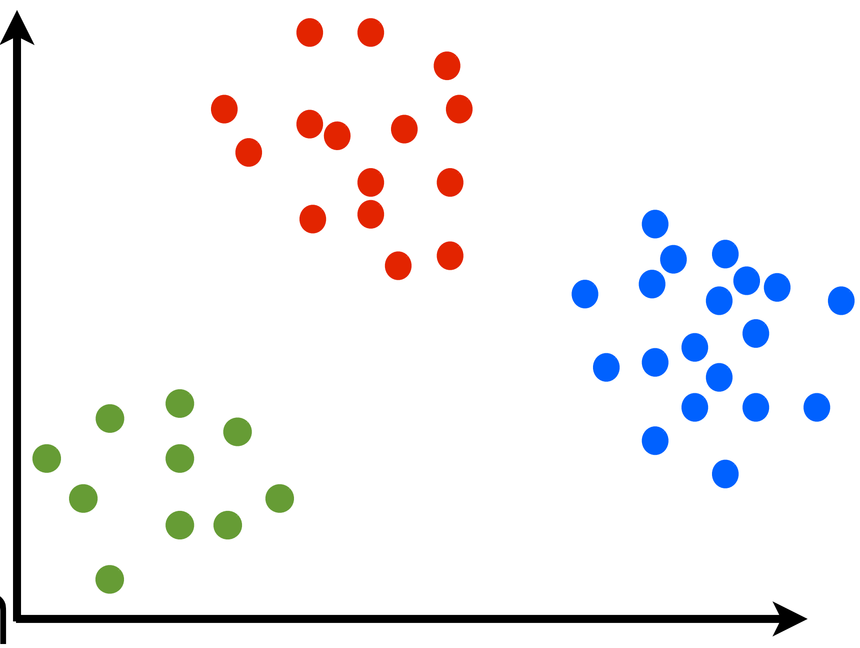


- A partition Π assigns the data to mutually exclusive & exhaustive groups
- Example problem: find a Bayesian estimate of the largest-cluster proportion
  - Let $h(\Pi)$ return proportion of data in the largest cluster
  - Let p_Π be the posterior on Π after observing the data
- Our general goal is to report: $H^* = \int h(\Pi) p_\Pi(\Pi) d\Pi$
- We often need to use a (random) approximation $\hat{H}$ for H^*
- Mean squared error (of the approx) = bias^2 + variance

$$\mathbb{E}[(\hat{H} - H^*)^2] = (\mathbb{E}\hat{H} - H^*)^2 + \mathrm{Var}(\hat{H})$$

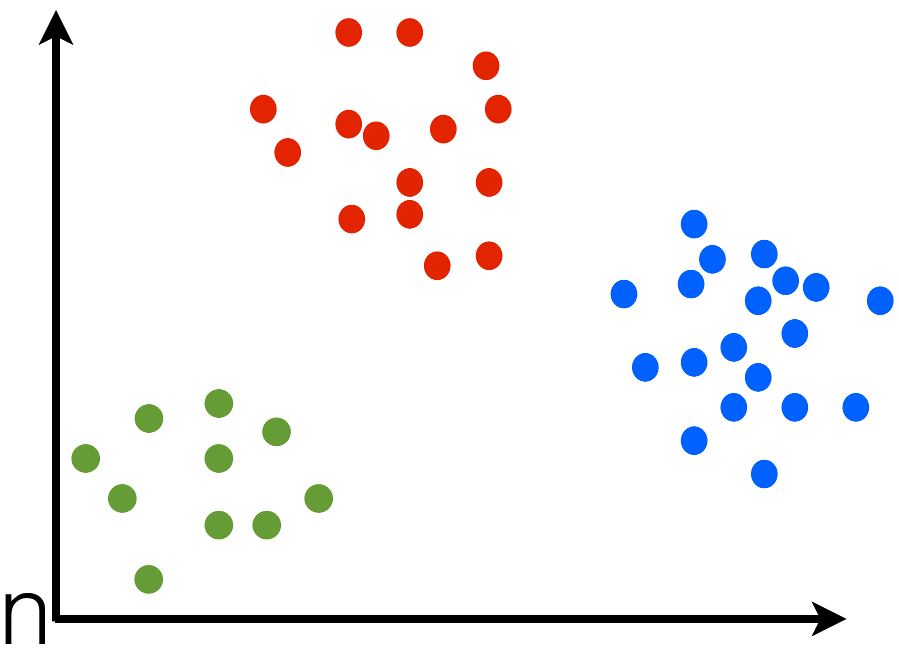
- Naive parallelism: replace $\hat{H}$ with an empirical average over many iid replicates → can reduce variance, not bias

Setup



- A partition Π assigns the data to mutually exclusive & exhaustive groups
- Example problem: find a Bayesian estimate of the largest-cluster proportion
  - Let $h(\Pi)$ return proportion of data in the largest cluster
  - Let p_Π be the posterior on Π after observing the data
- Our general goal is to report: $H^* = \int h(\Pi) p_\Pi(\Pi) d\Pi$
- We often need to use a (random) approximation $\hat{H}$ for H^*
- Mean squared error (of the approx) = bias² + variance
$$\mathbb{E}[(\hat{H} - H^*)^2] = (\mathbb{E}\hat{H} - H^*)^2 + \mathrm{Var}(\hat{H})$$
- Naive parallelism: replace $\hat{H}$ with an empirical average over many iid replicates → can reduce variance, not bias
  - If $\hat{H}$ unbiased, can reach any small error with more cores

Setup



- A partition Π assigns the data to mutually exclusive & exhaustive groups
- Example problem: find a Bayesian estimate of the largest-cluster proportion
  - Let $h(\Pi)$ return proportion of data in the largest cluster
  - Let p_Π be the posterior on Π after observing the data
- Our general goal is to report: $H^* = \int h(\Pi) p_\Pi(\Pi) d\Pi$
- We often need to use a (random) approximation $\hat{H}$ for H^*
- Mean squared error (of the approx) = bias² + variance

$$\mathbb{E}[(\hat{H} - H^*)^2] = (\mathbb{E}\hat{H} - H^*)^2 + \mathrm{Var}(\hat{H})$$

- Naive parallelism: replace $\hat{H}$ with an empirical average over many iid replicates → can reduce variance, not bias
  - If $\hat{H}$ unbiased, can reach any small error with more cores
    - Historical aside: unbiasedness → bias is fine → unbiasedness

Coupling for removing bias

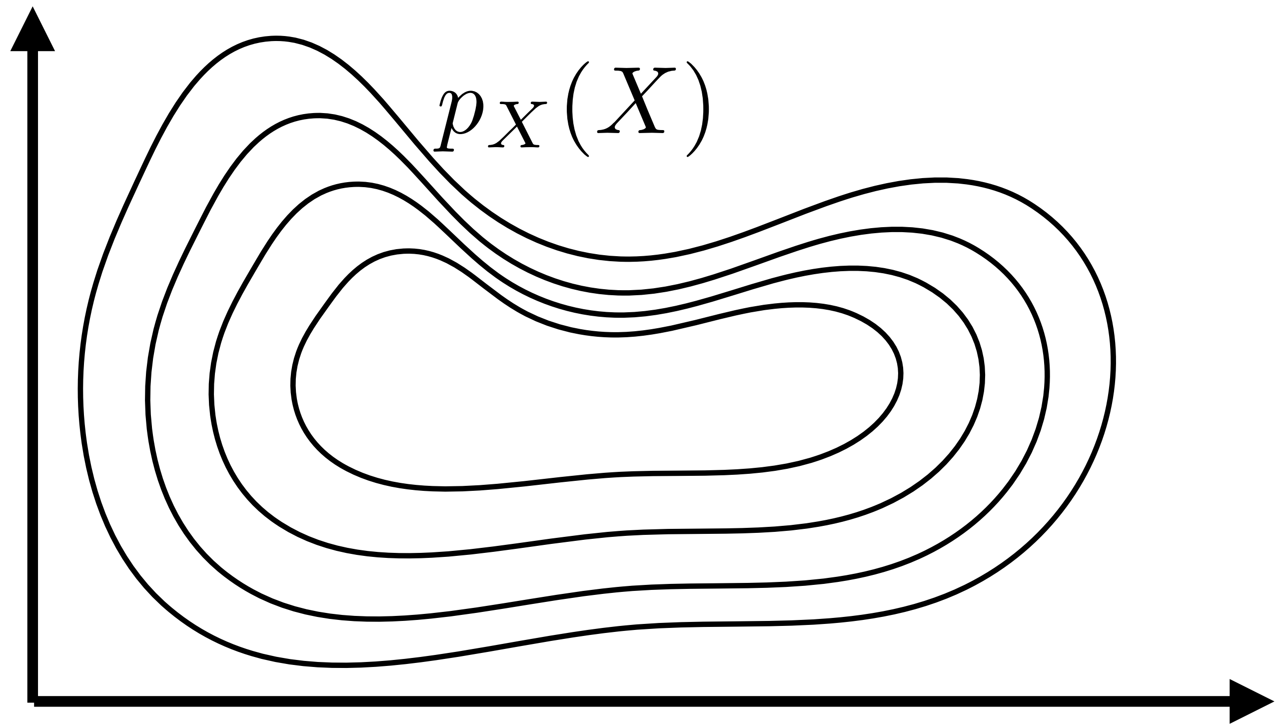
- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion

Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo:

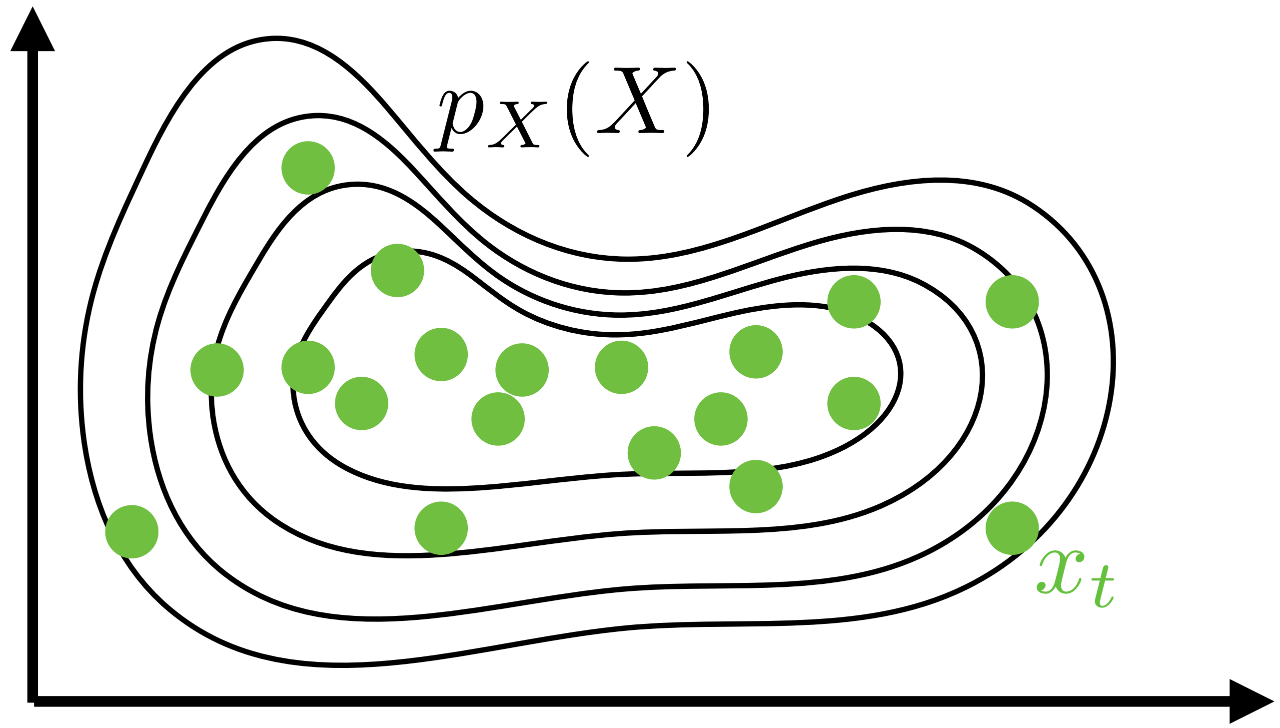
Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo:



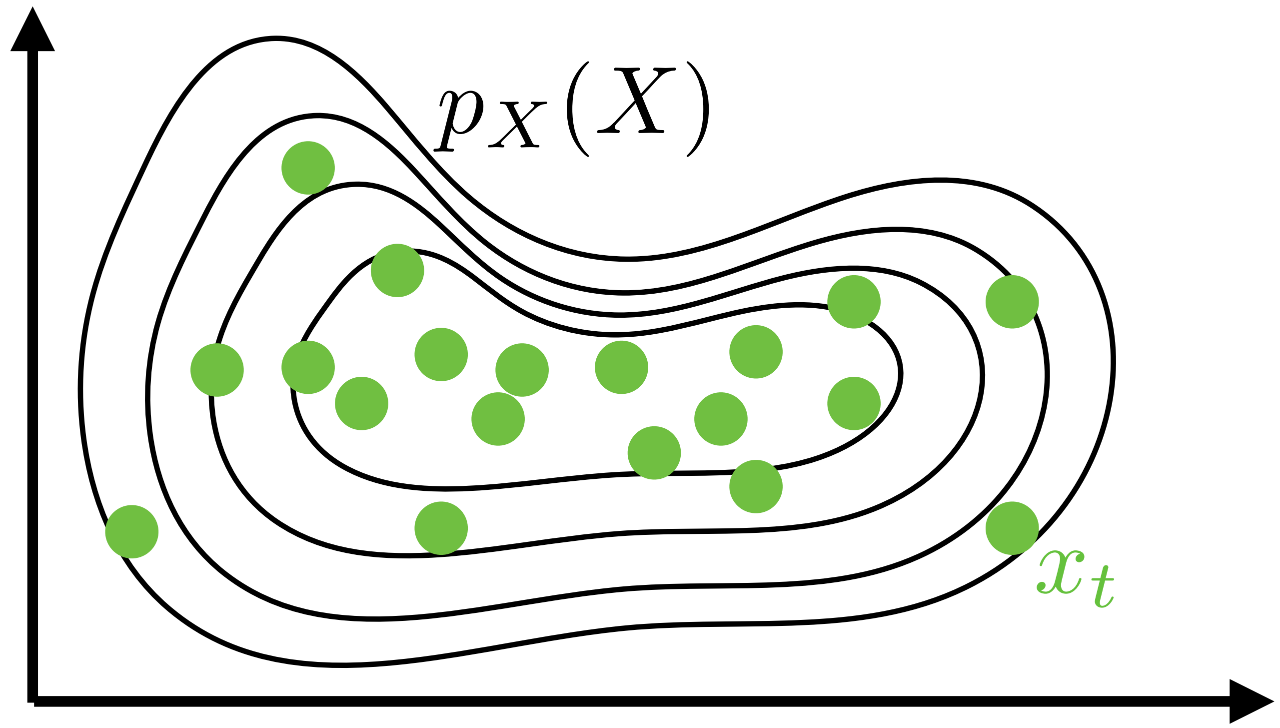
Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo:



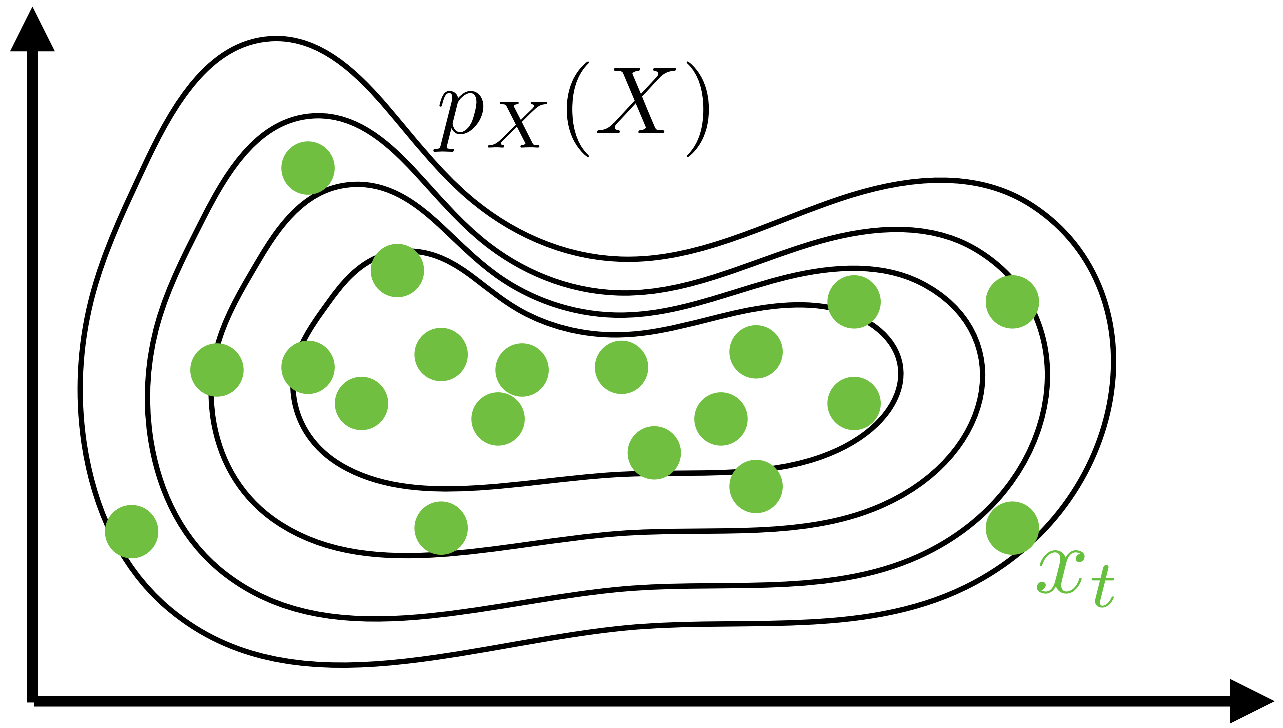
Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: → $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$



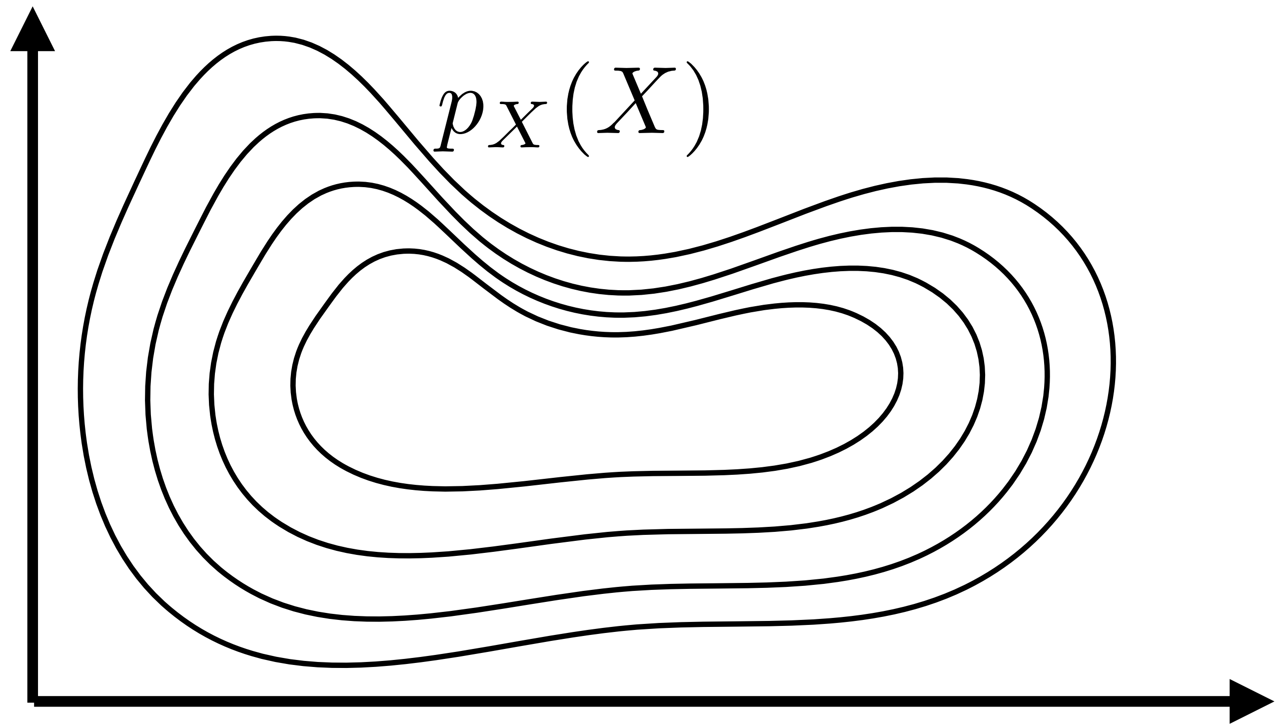
Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$



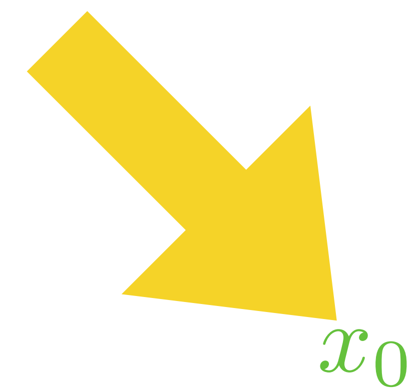
Coupling for removing bias

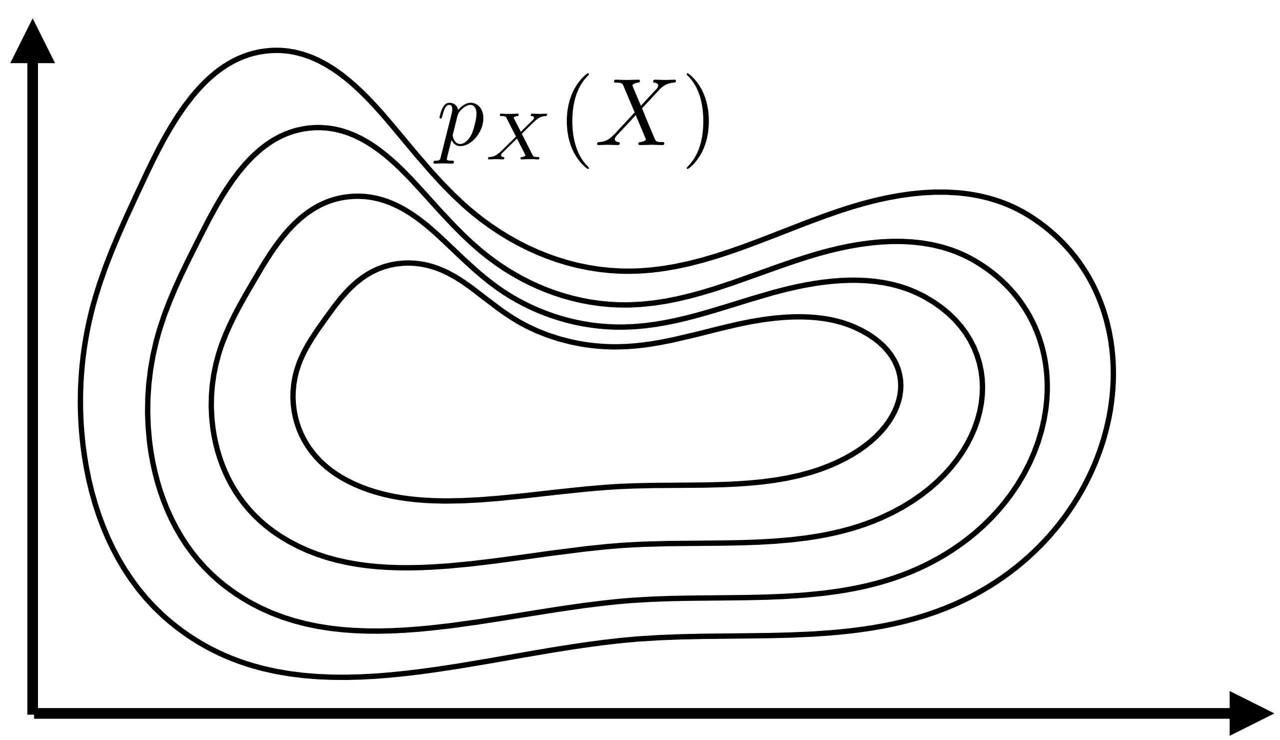
- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible ⟶ $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo



Coupling for removing bias

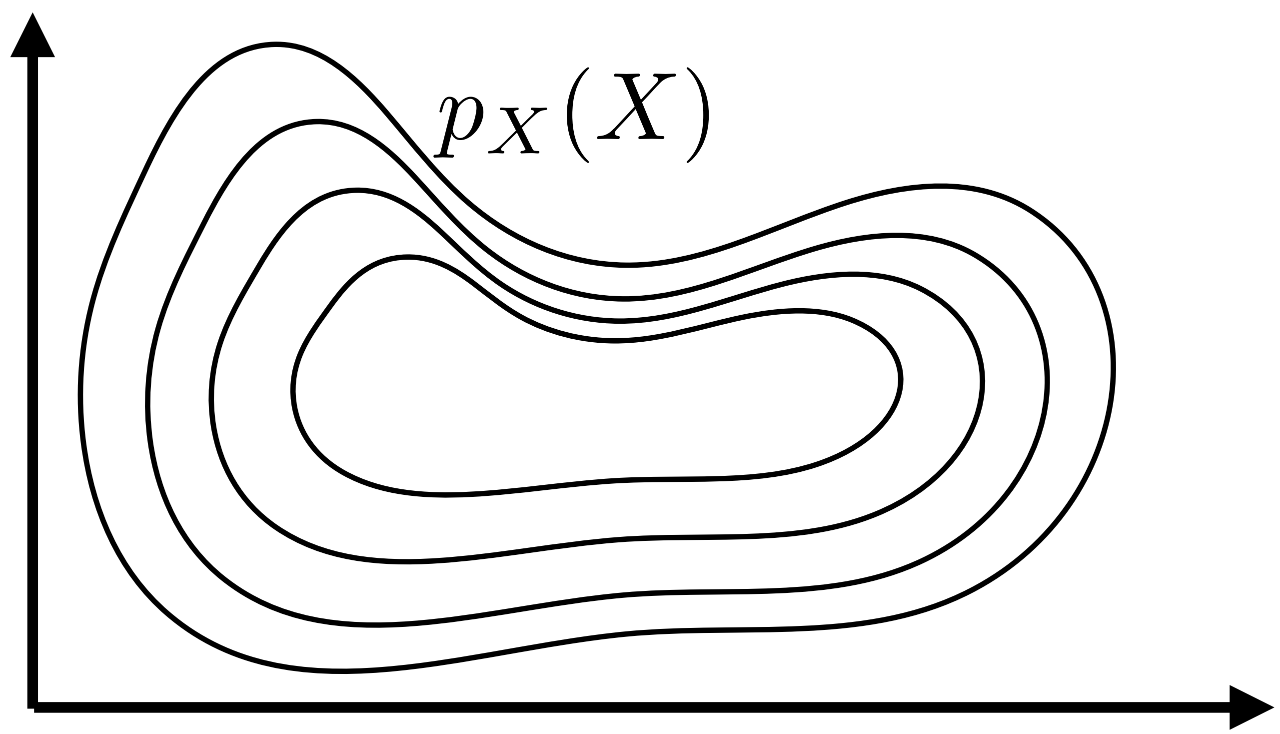
- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo

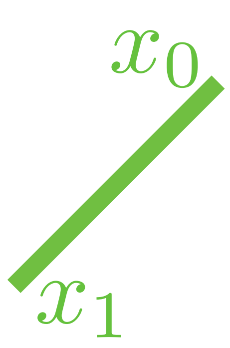




Coupling for removing bias

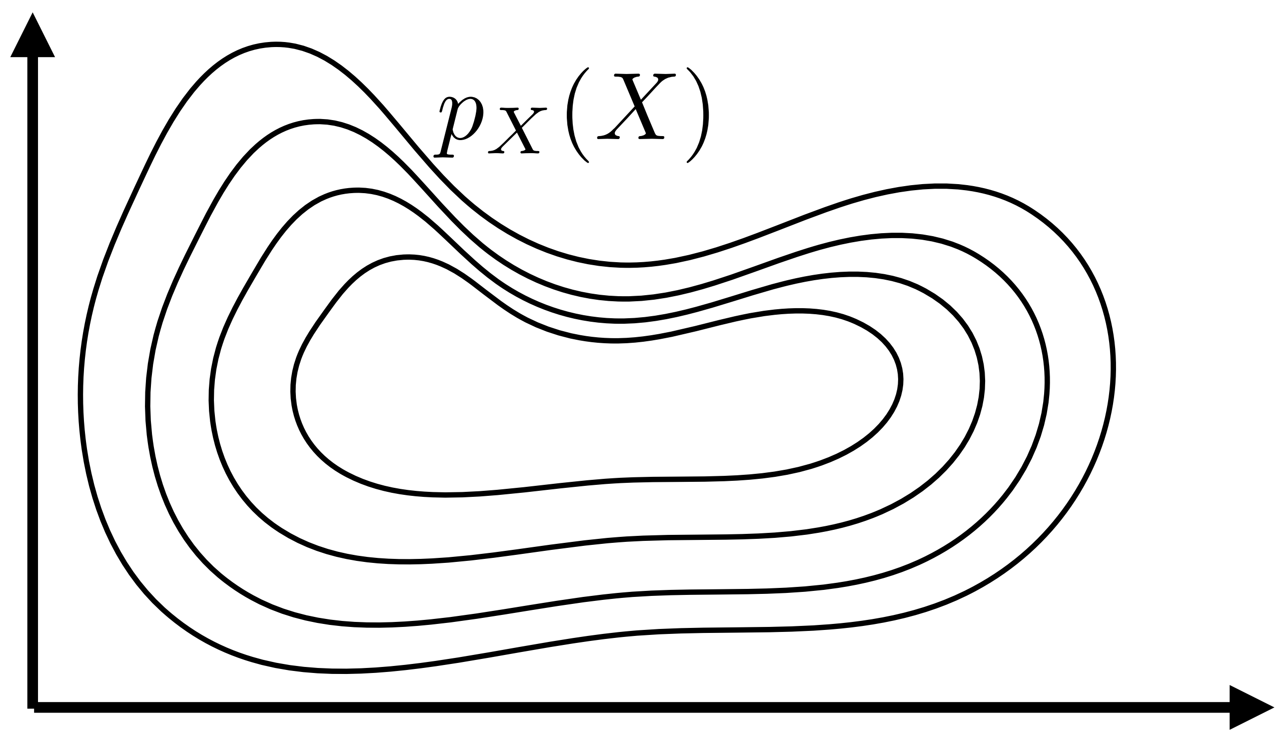
- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible $\longrightarrow$ $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo

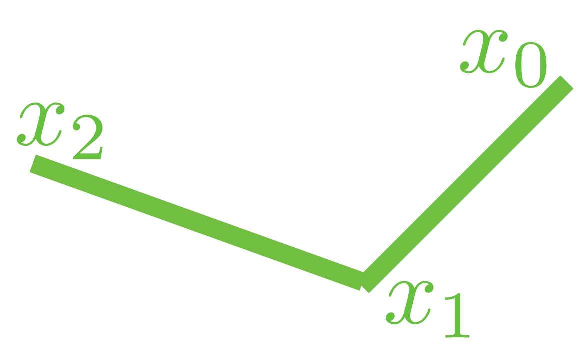




Coupling for removing bias

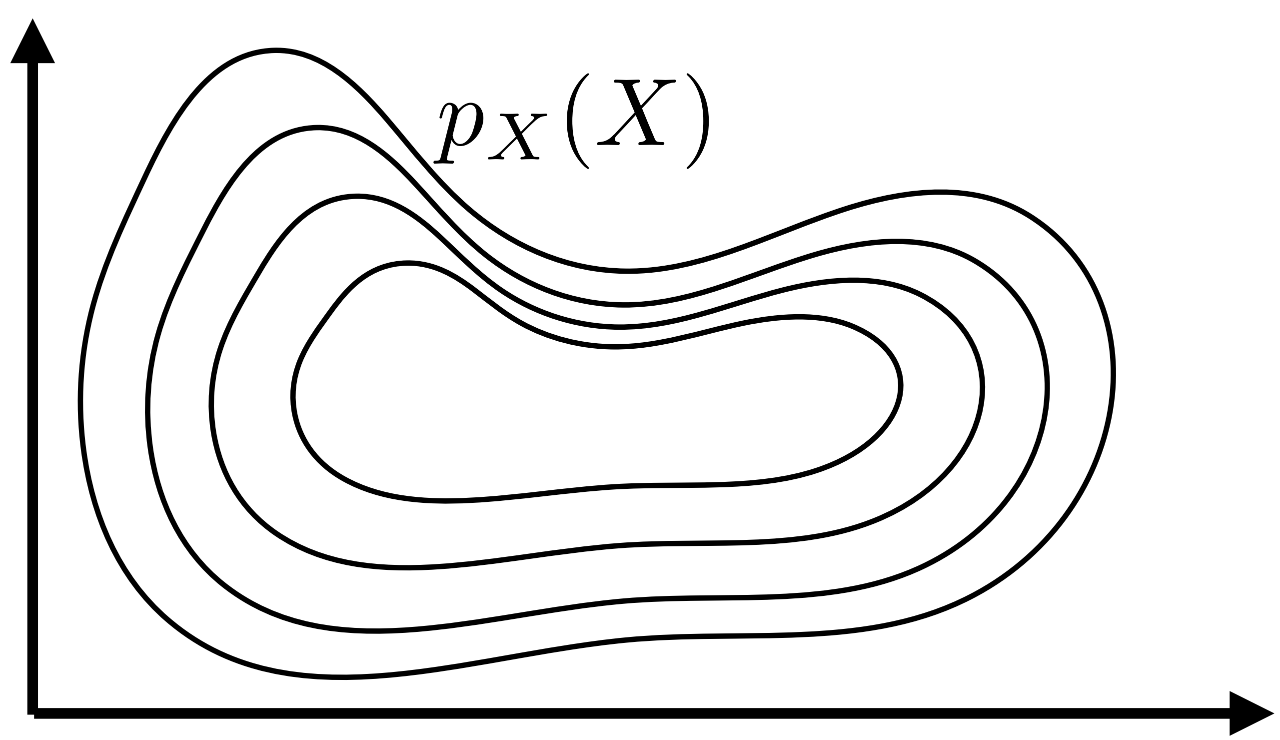
- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible ⟶ $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo

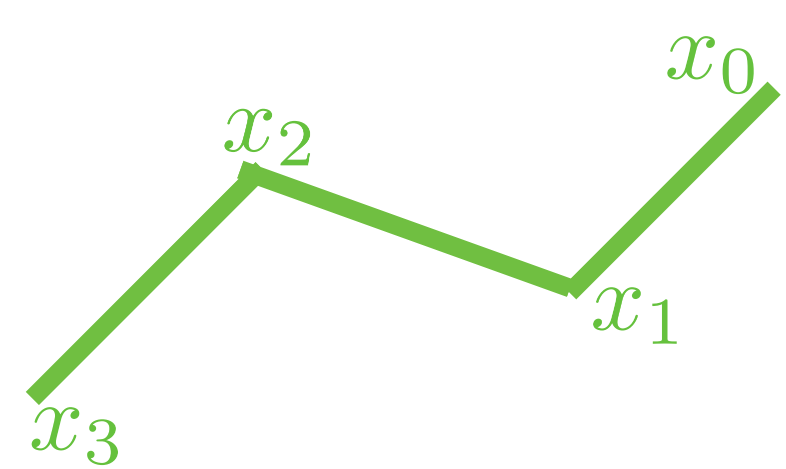




Coupling for removing bias

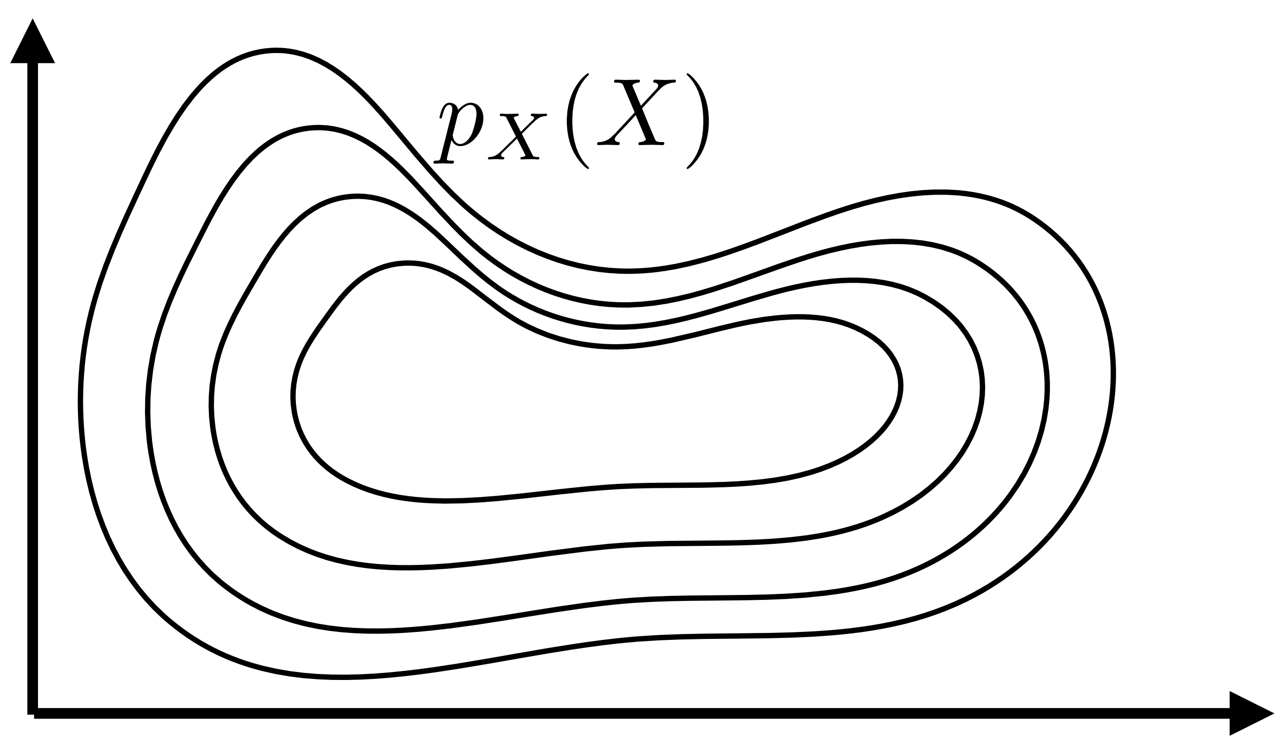
- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible $\longrightarrow$ $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo

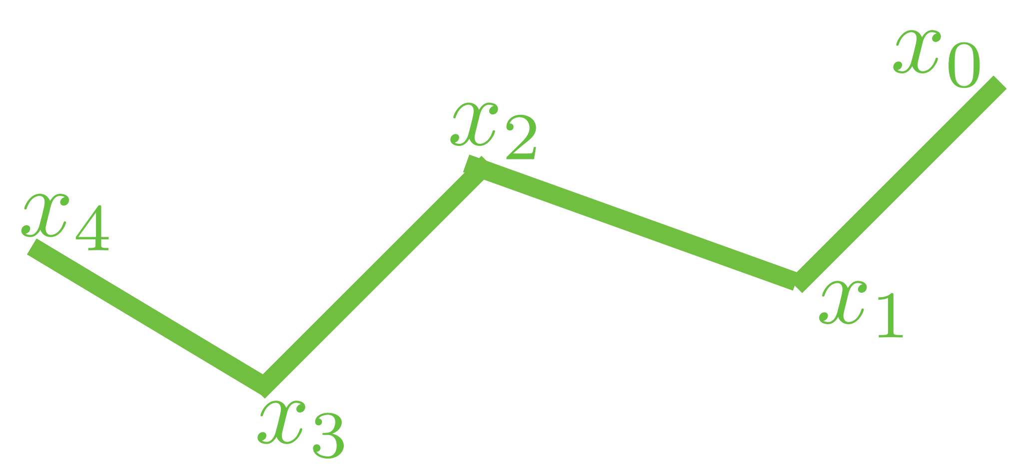




Coupling for removing bias

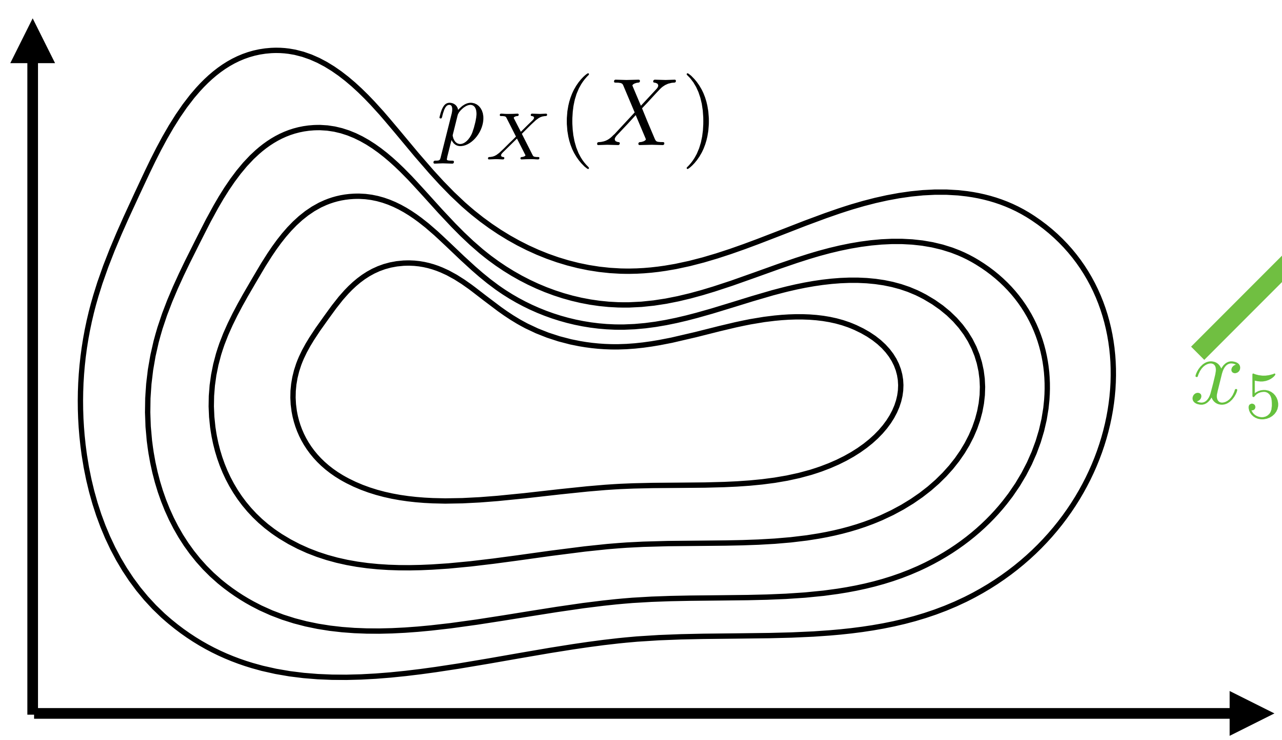
- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible $\longrightarrow \hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo

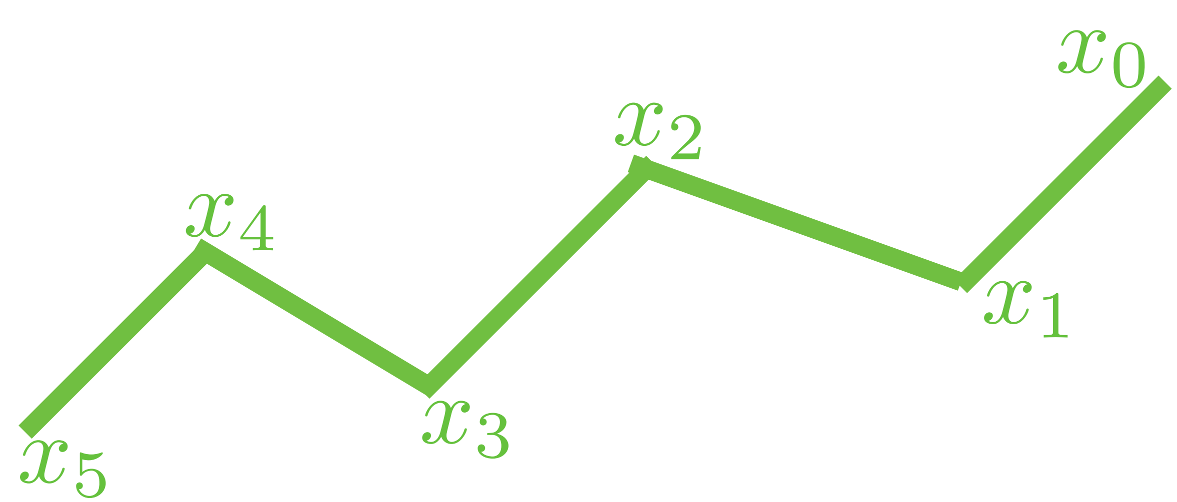




Coupling for removing bias

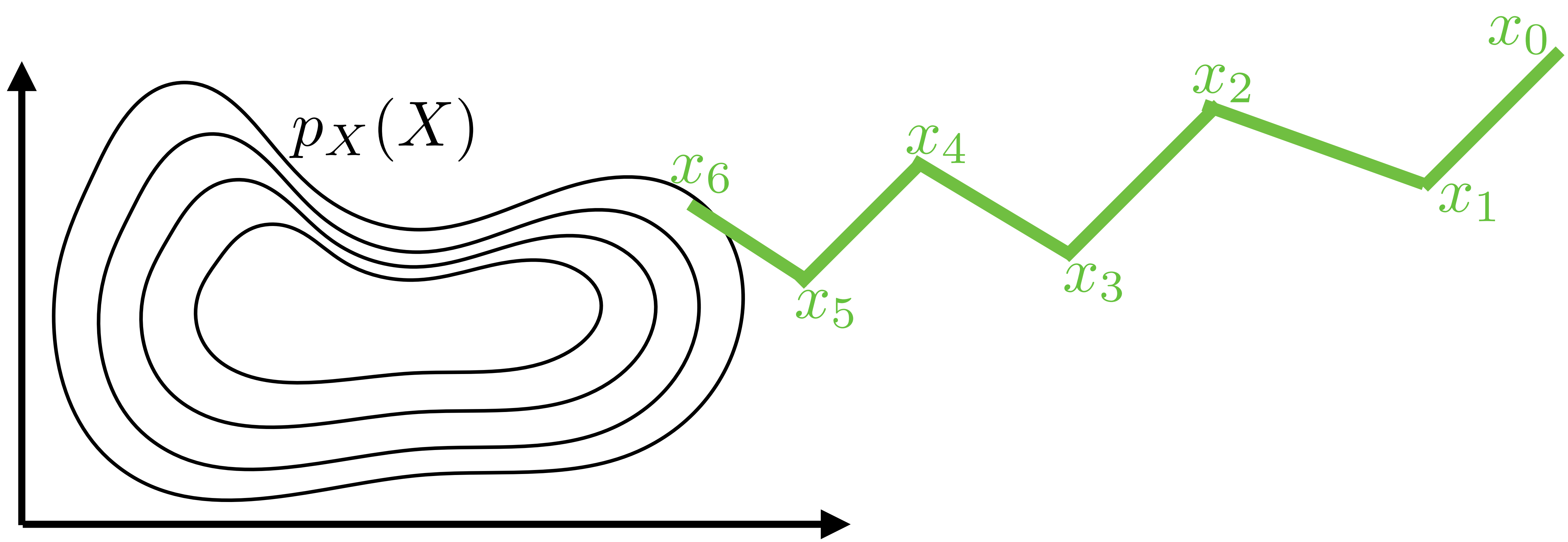
- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
    - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible ⟶ $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo





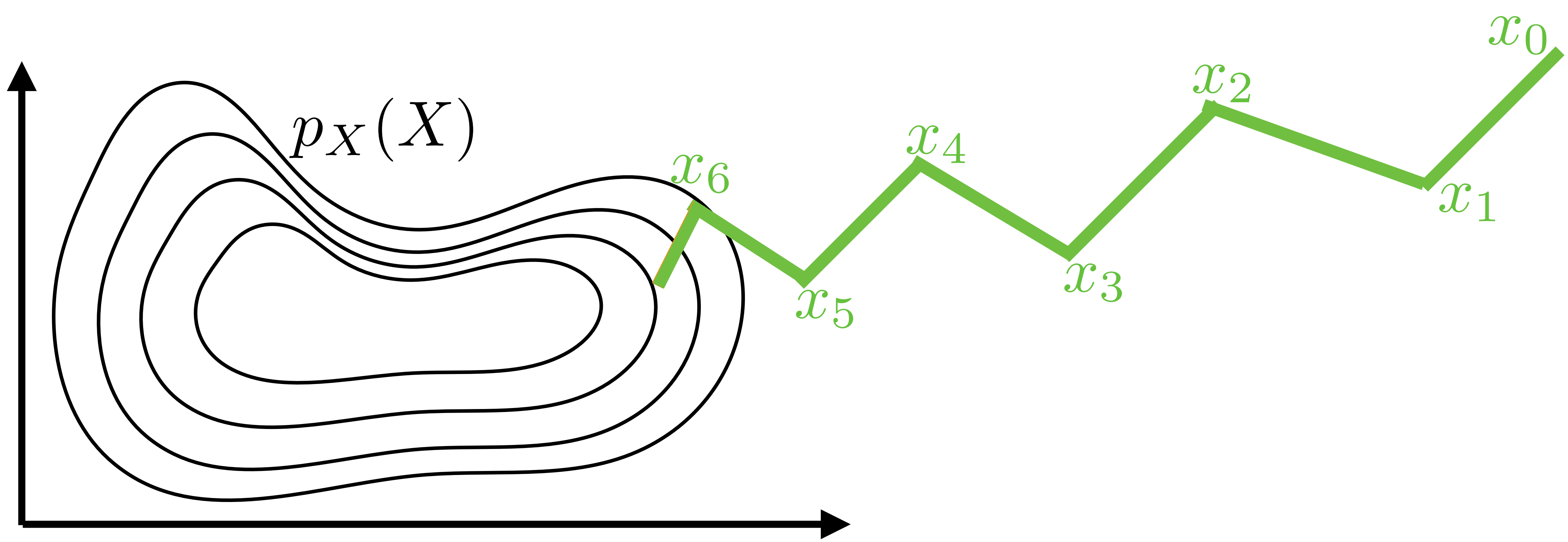
Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
    - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible ⟶ $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo



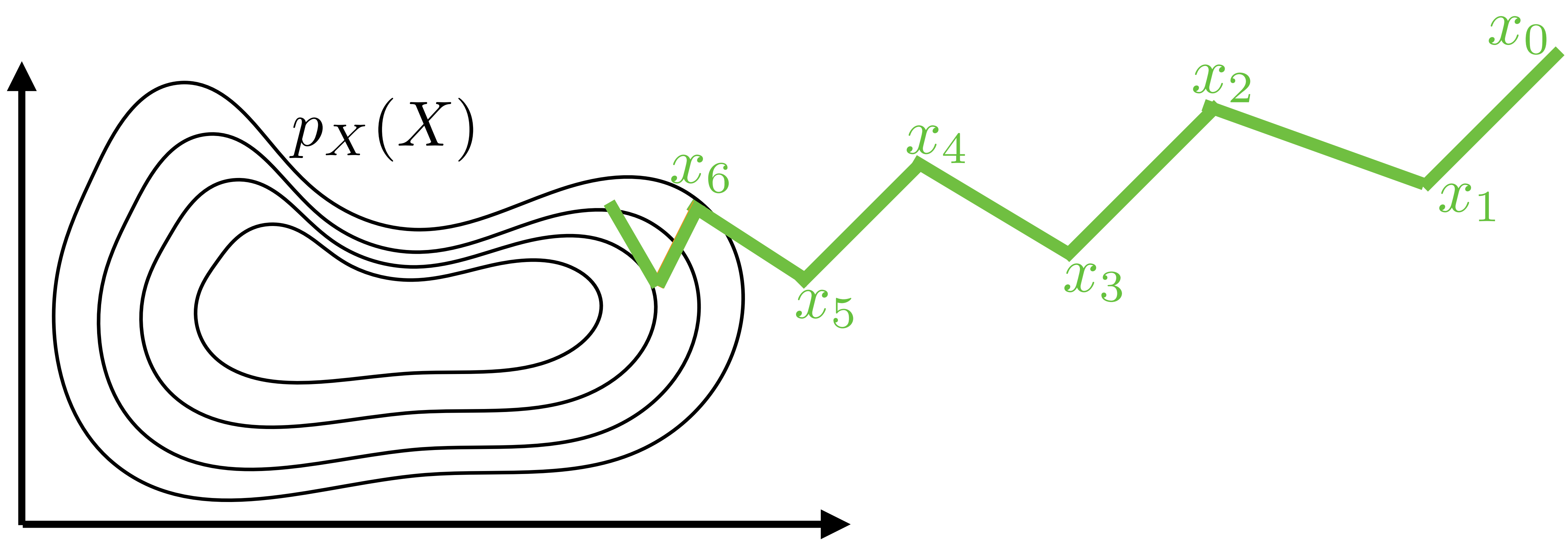
Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible ⟶ $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo



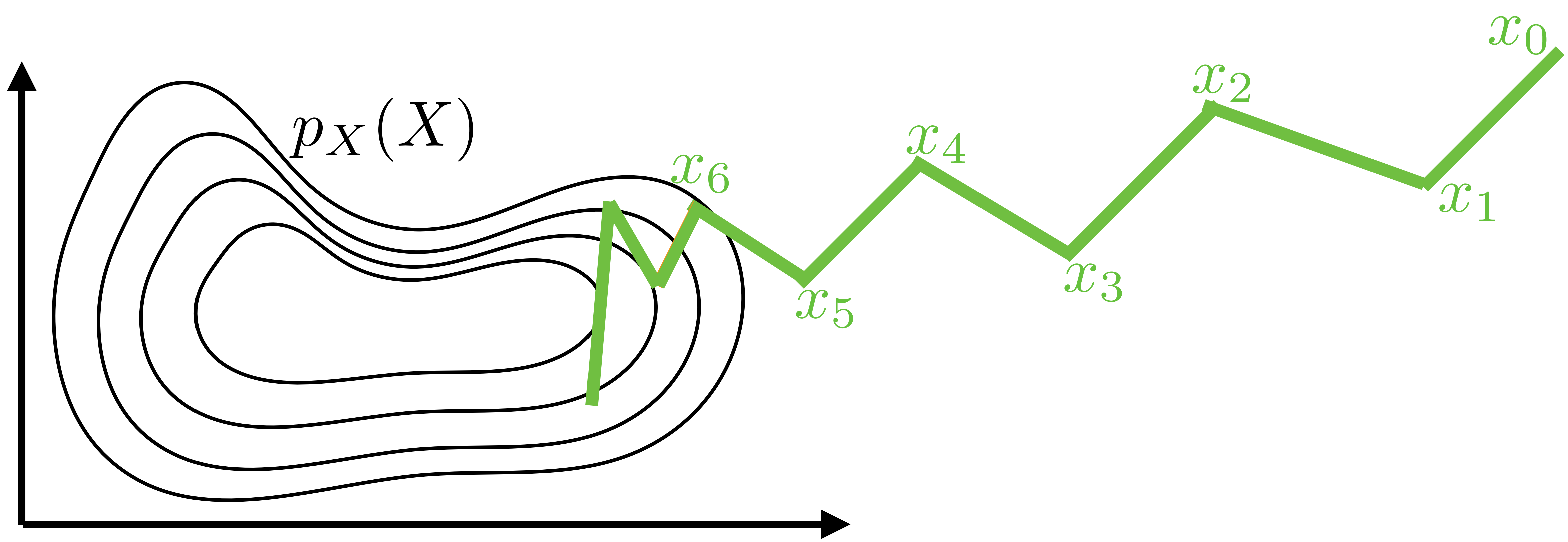
Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible ⟶ $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo



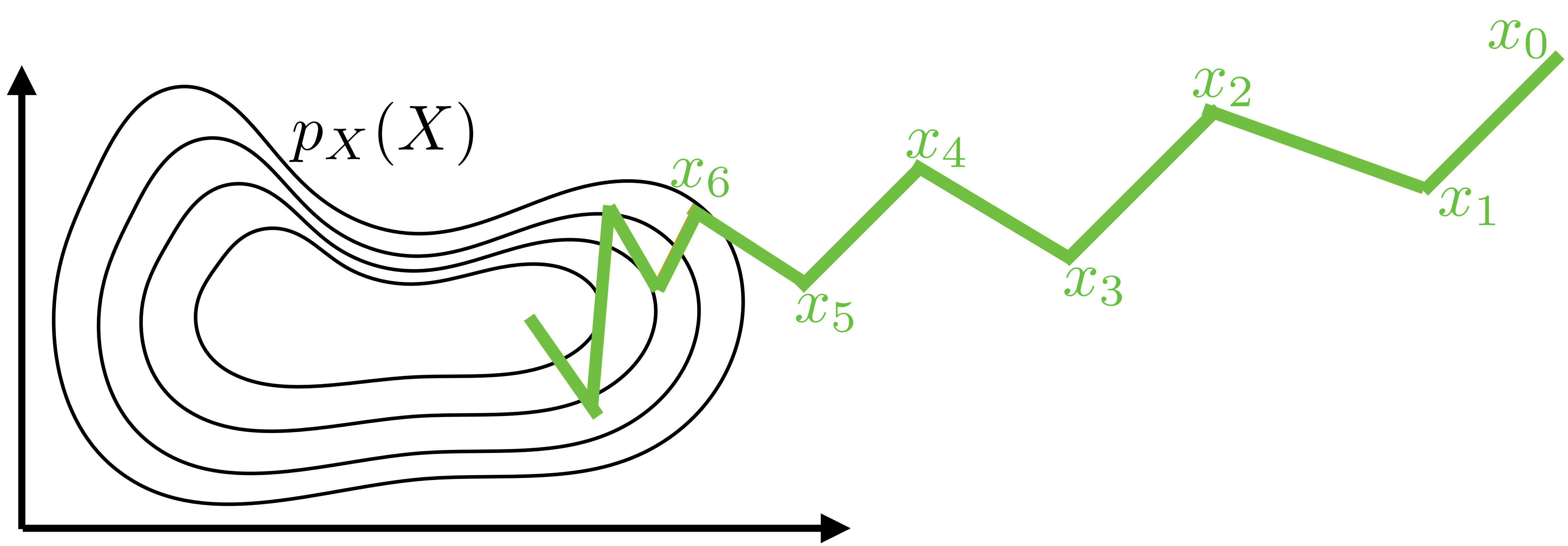
Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible ⟶ $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo



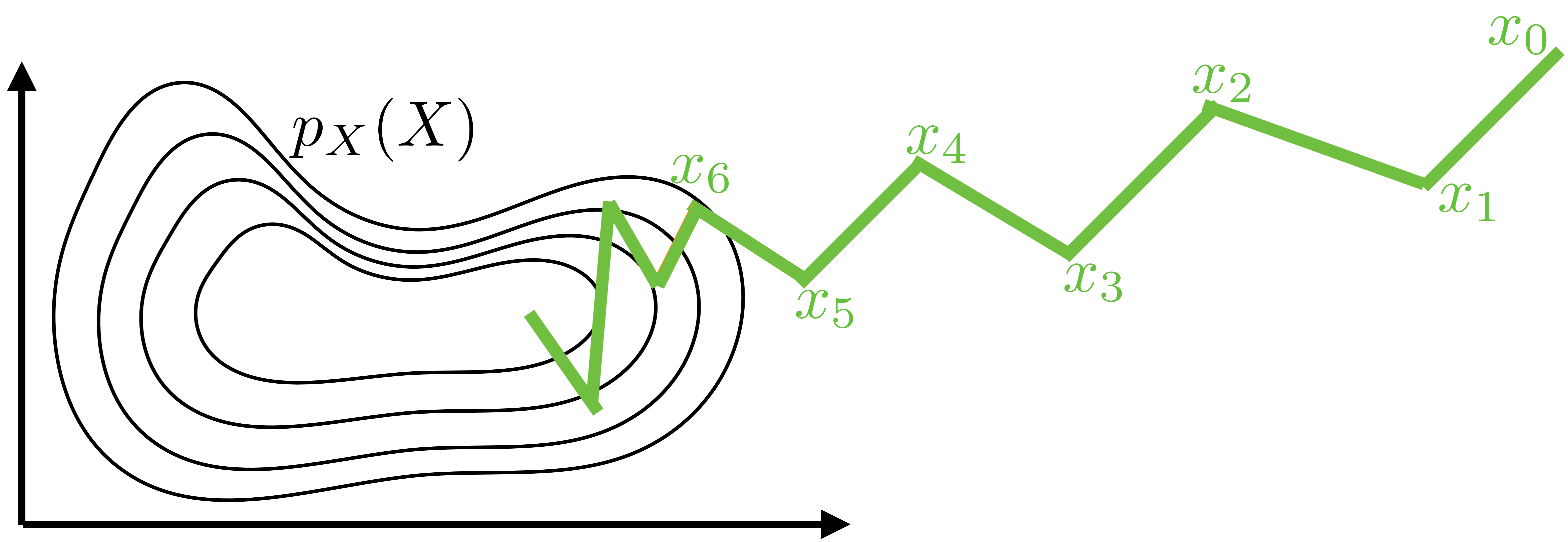
Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible $\longrightarrow$ $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo



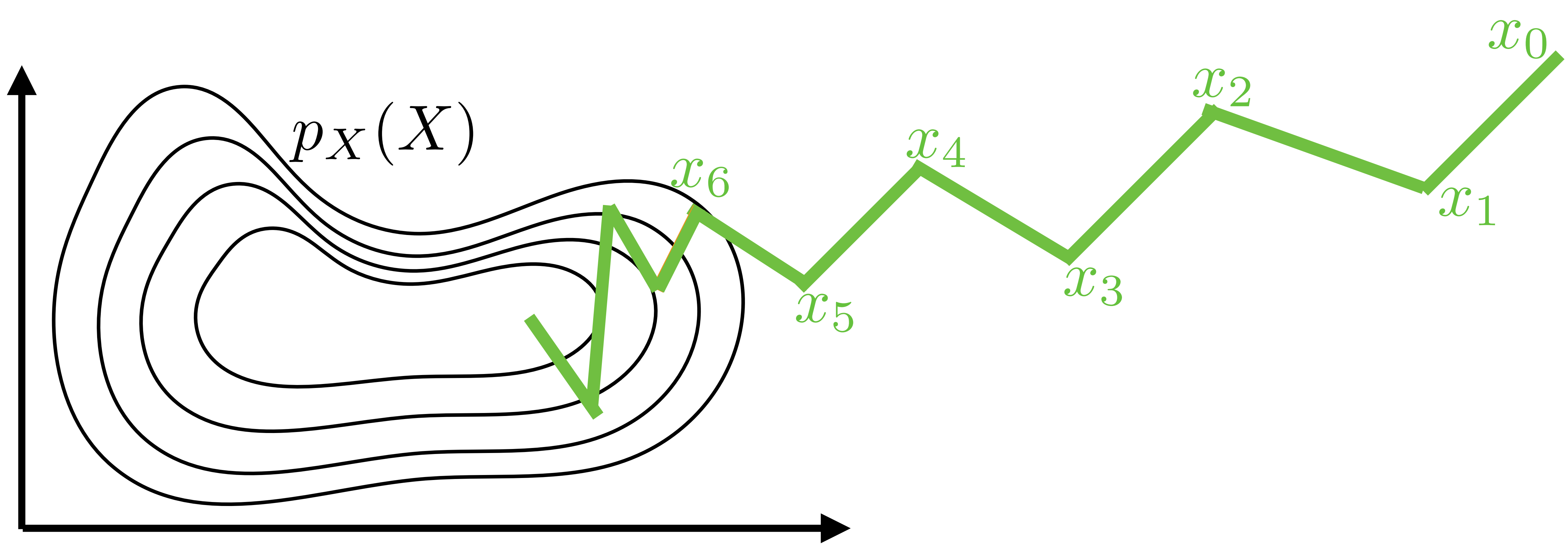
Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo



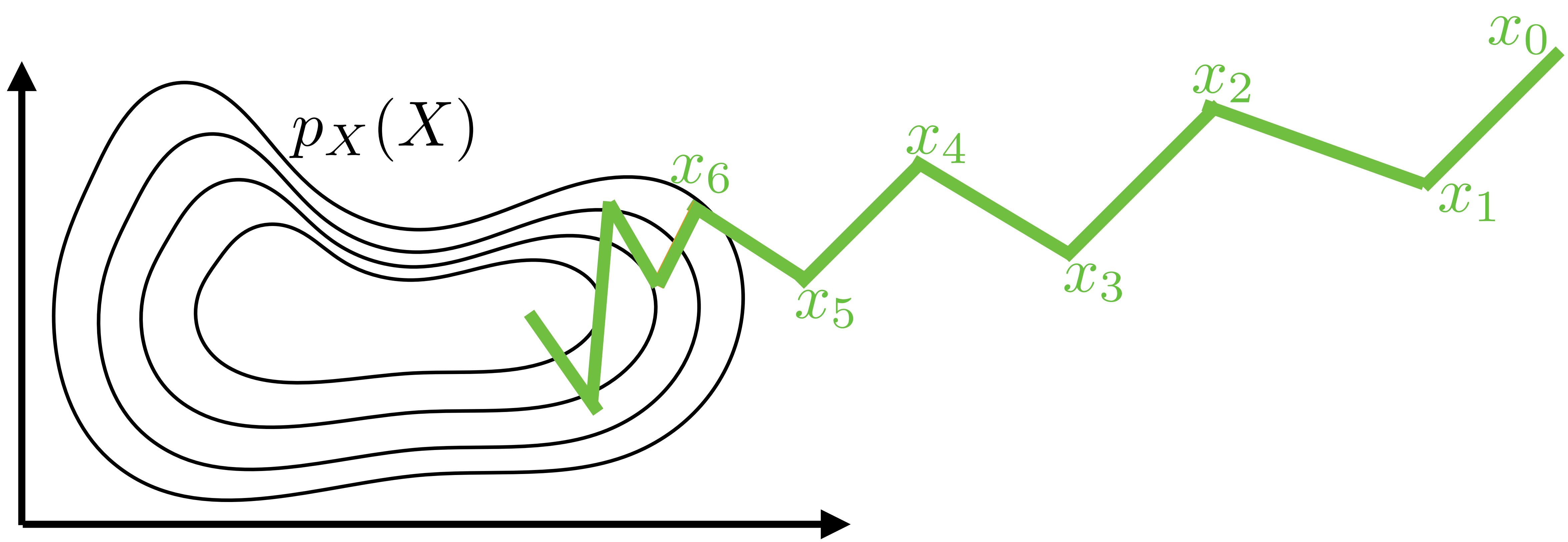
Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time



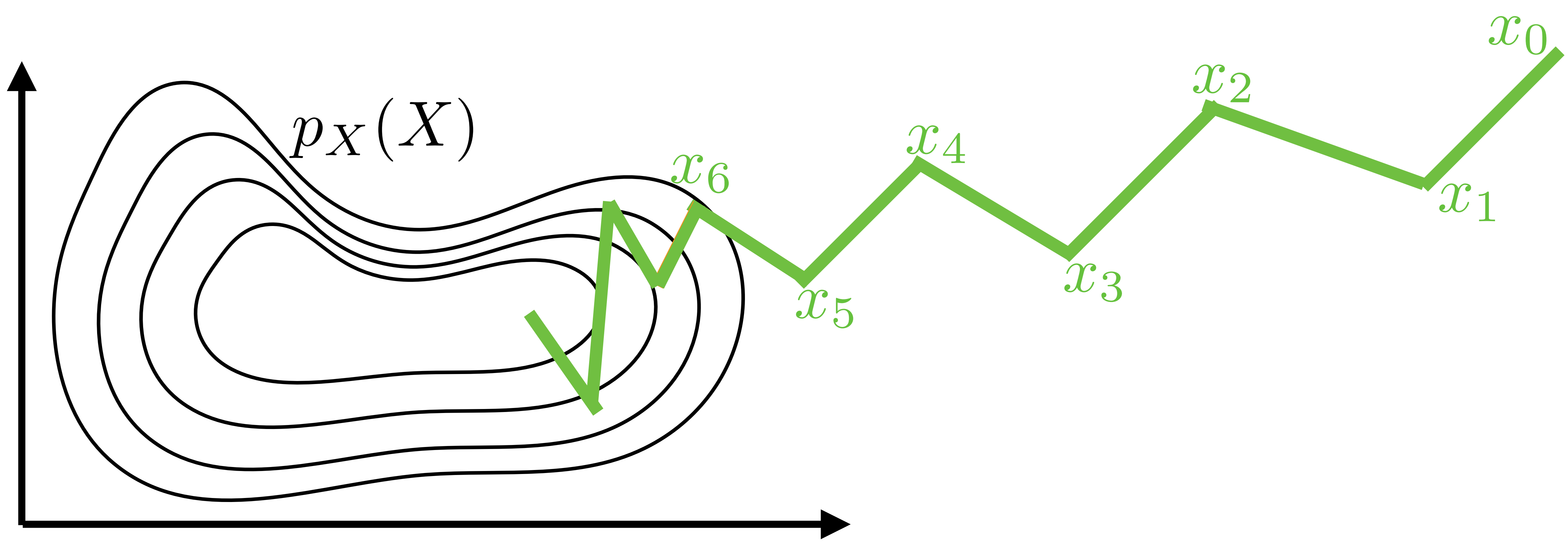
Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time
- Coupling [Jacob et al 2020]



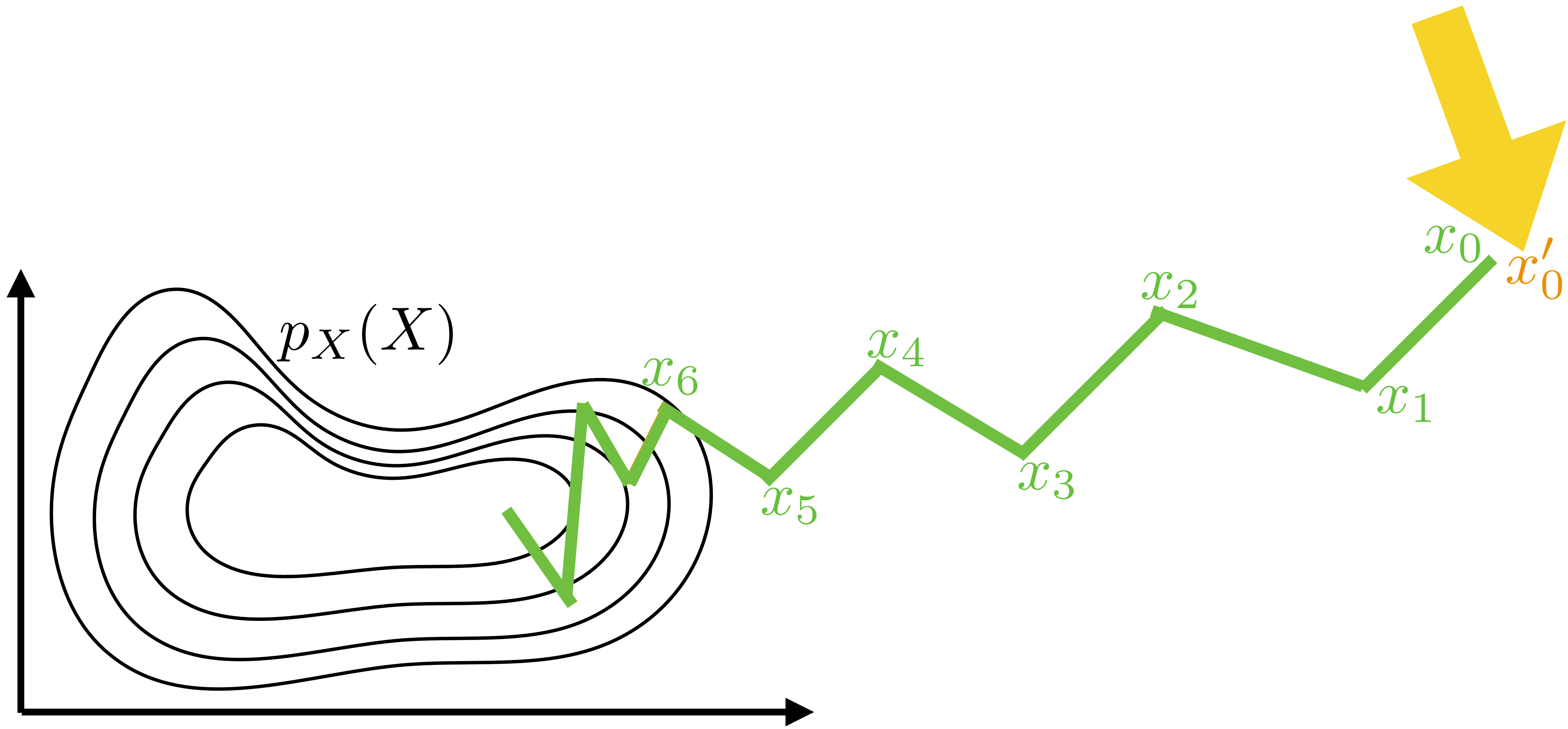
Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T}\sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time
- Coupling [Jacob et al 2020]: make two MCMC chains



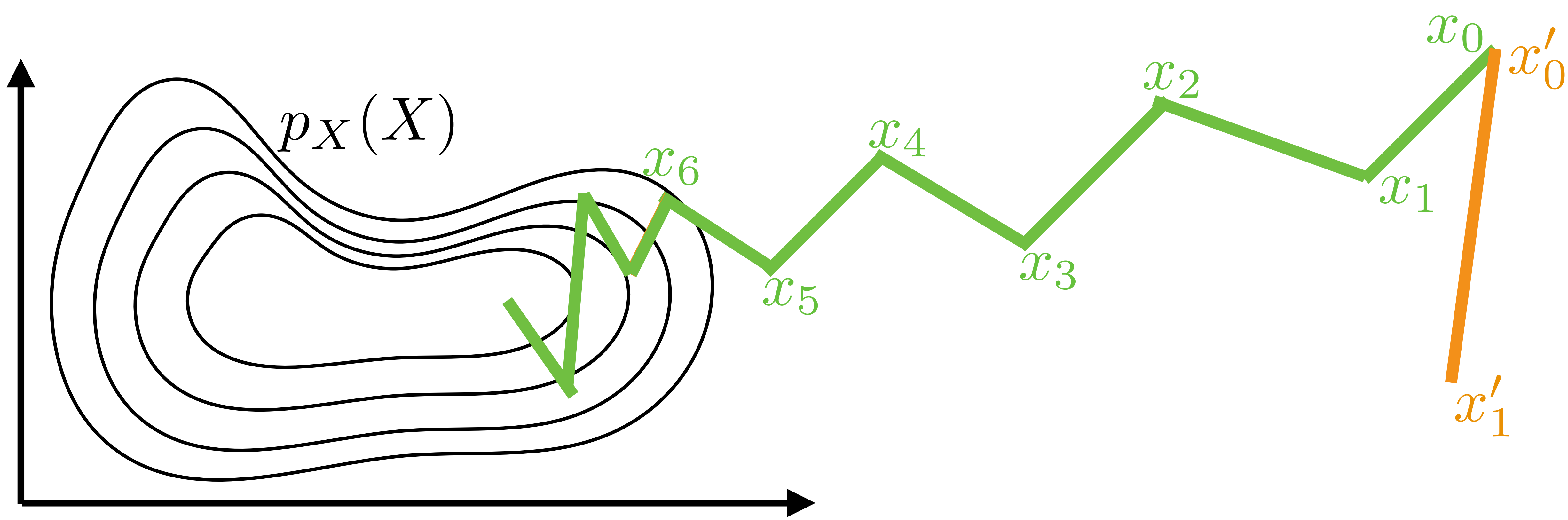
Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time
- Coupling [Jacob et al 2020]: make two MCMC chains



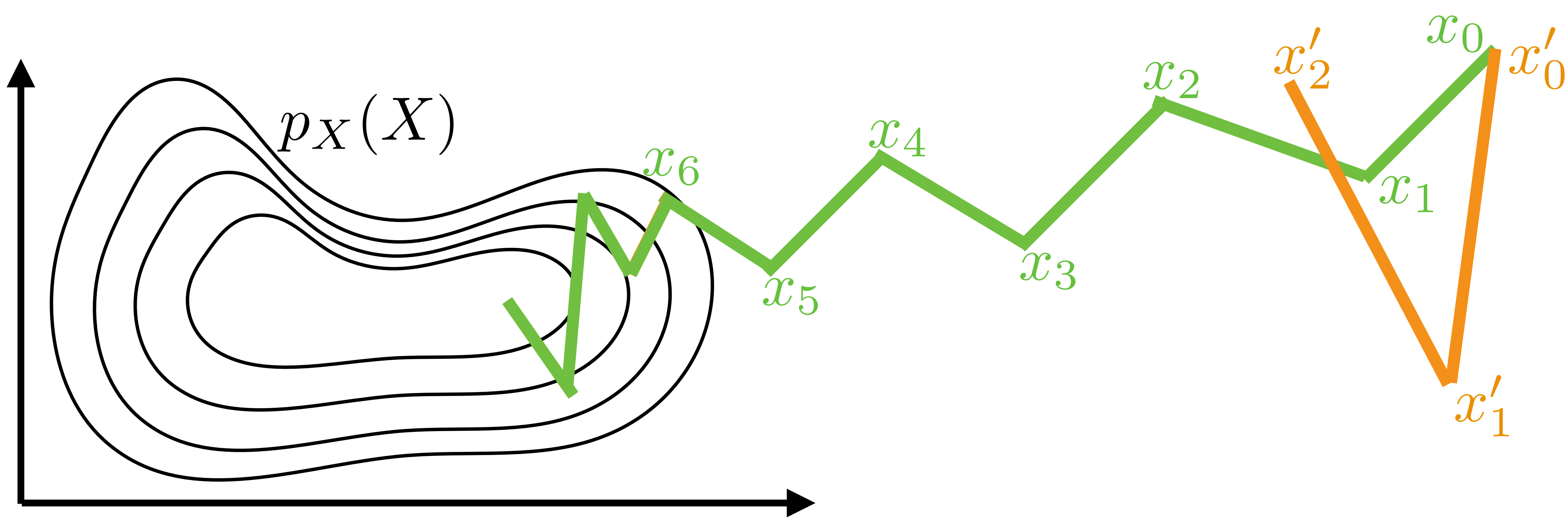
Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time
- Coupling [Jacob et al 2020]: make two MCMC chains



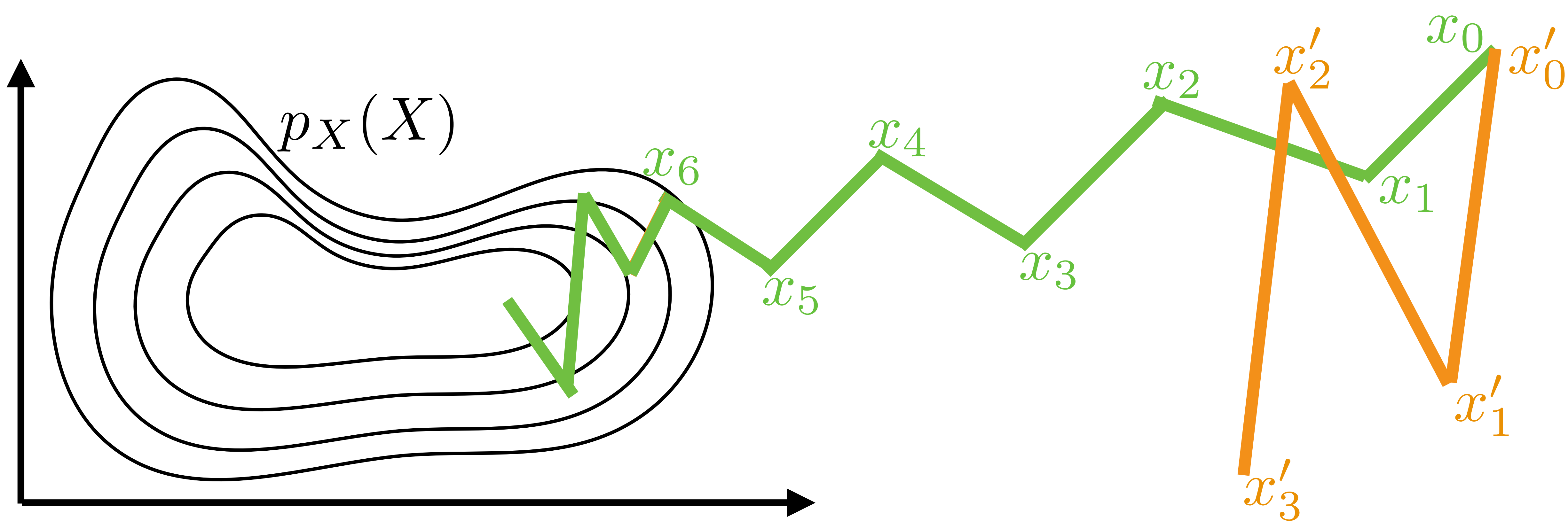
Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X)p_X(X)dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T}\sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time
- Coupling [Jacob et al 2020]: make two MCMC chains



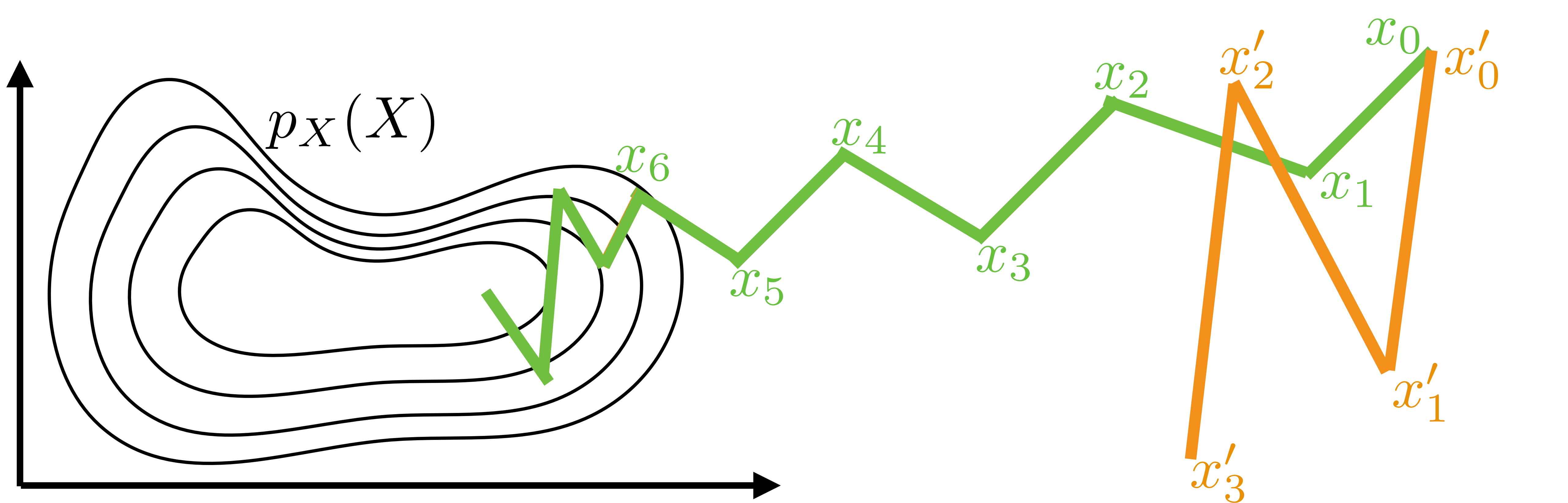
Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T}\sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time
- Coupling [Jacob et al 2020]: make two MCMC chains



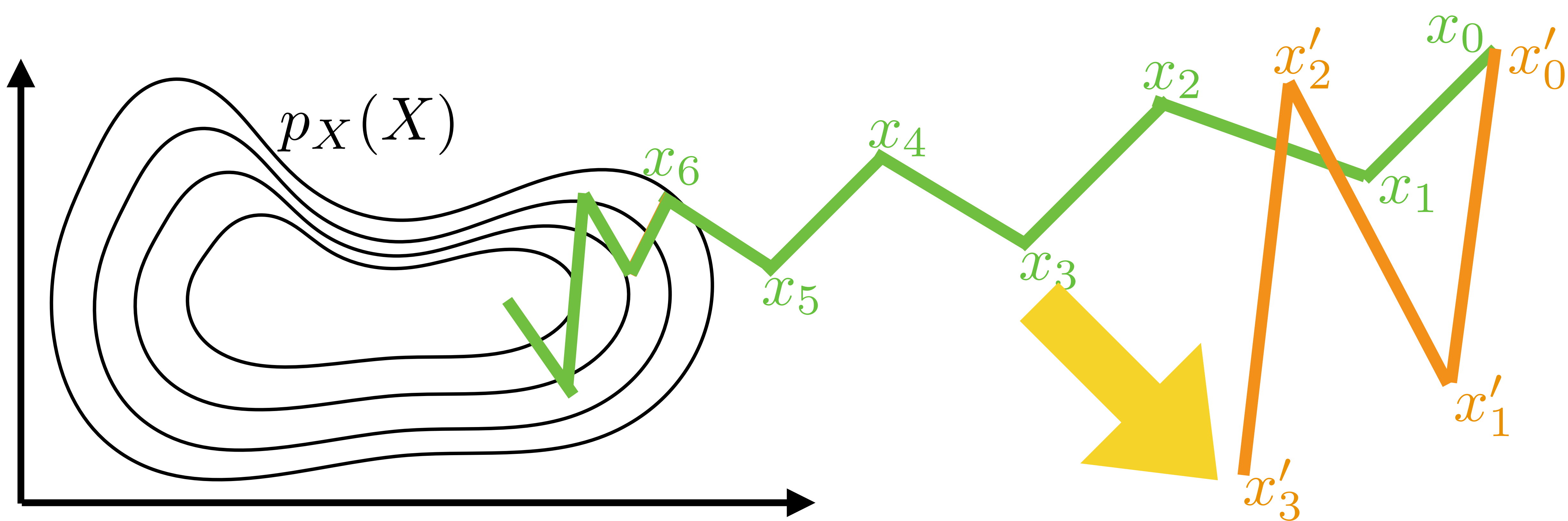
Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time
- Coupling [Jacob et al 2020]: make two MCMC chains; set up joint dynamics so chains meet exactly & quickly



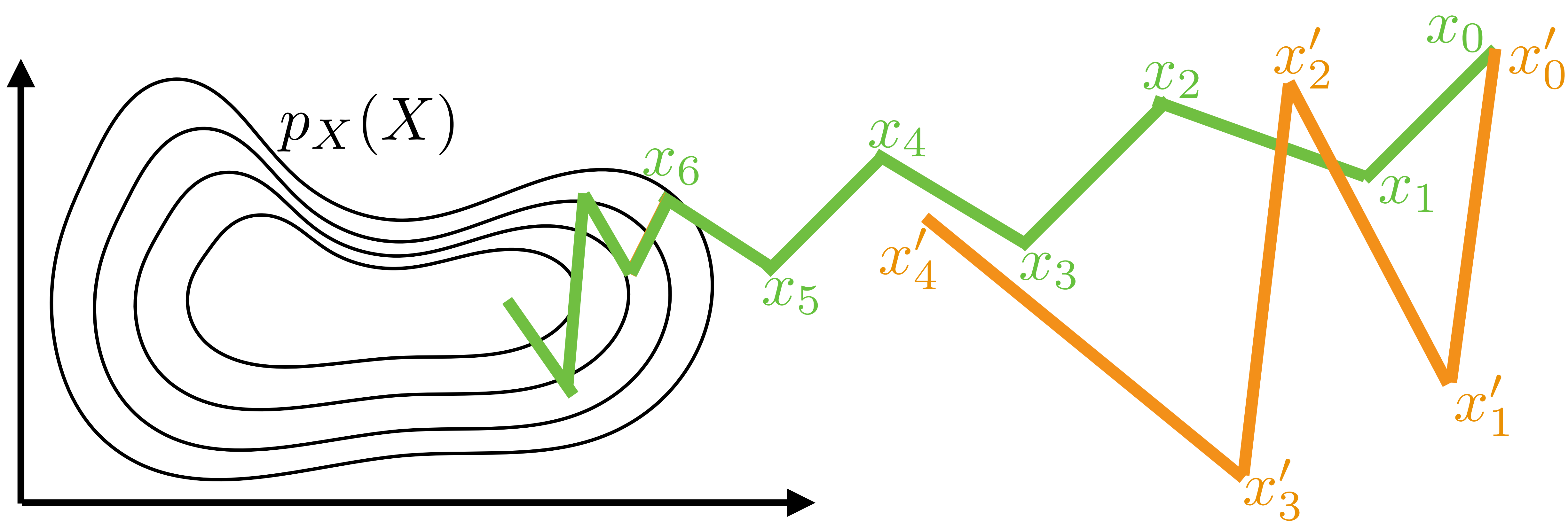
Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time
- Coupling [Jacob et al 2020]: make two MCMC chains; set up joint dynamics so chains meet exactly & quickly



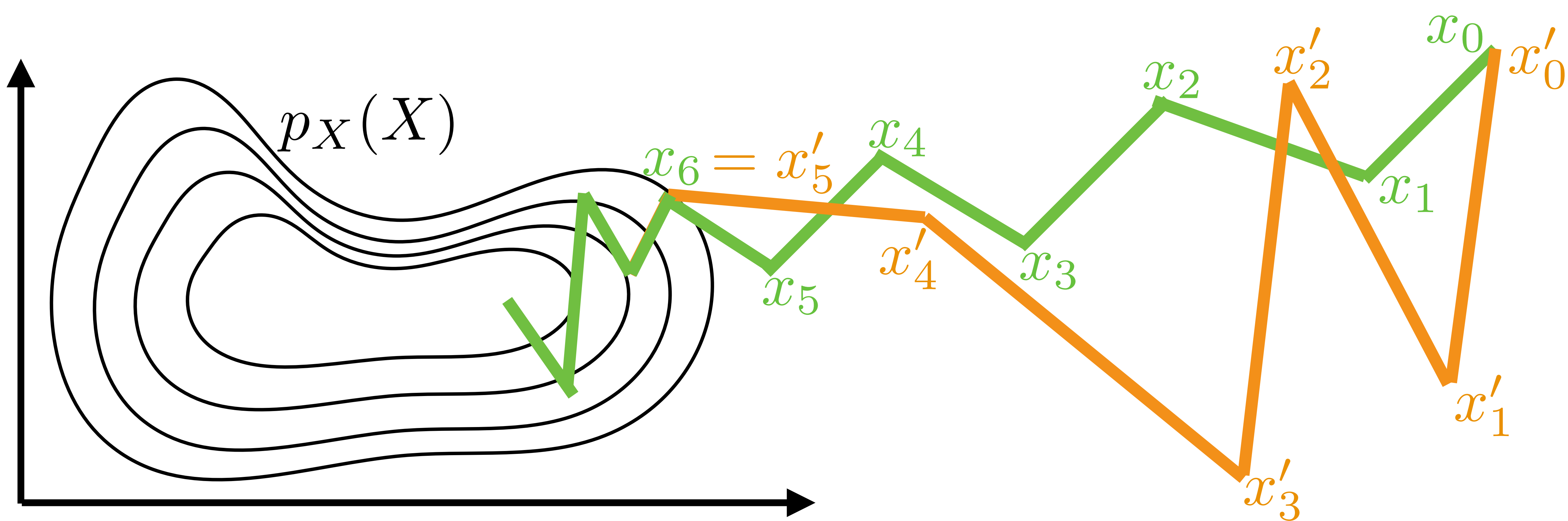
Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
    - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time
- Coupling [Jacob et al 2020]: make two MCMC chains; set up joint dynamics so chains meet exactly & quickly



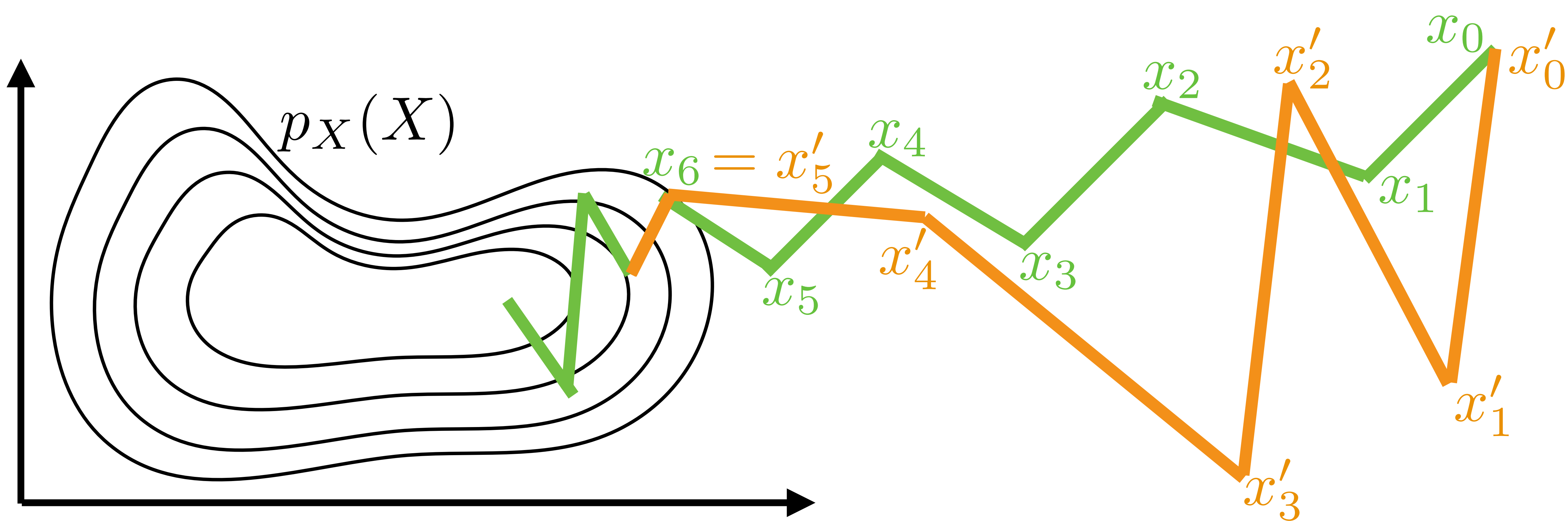
Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time
- Coupling [Jacob et al 2020]: make two MCMC chains; set up joint dynamics so chains meet exactly & quickly



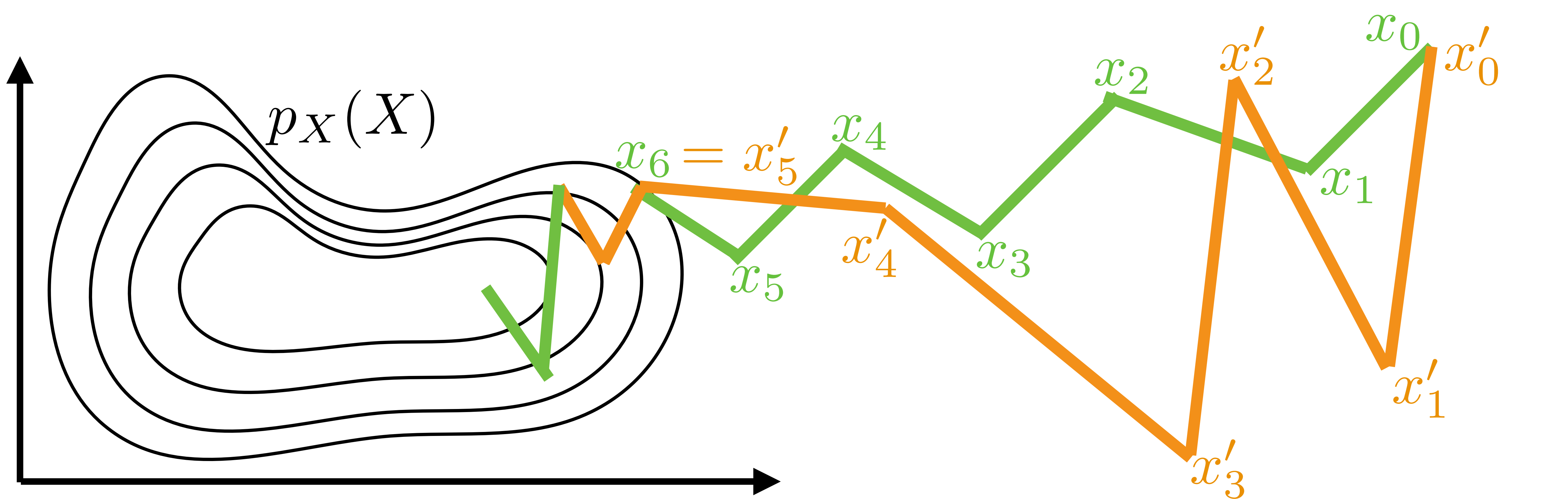
Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible $\longrightarrow \hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time
- Coupling [Jacob et al 2020]: make two MCMC chains; set up joint dynamics so chains meet exactly & quickly



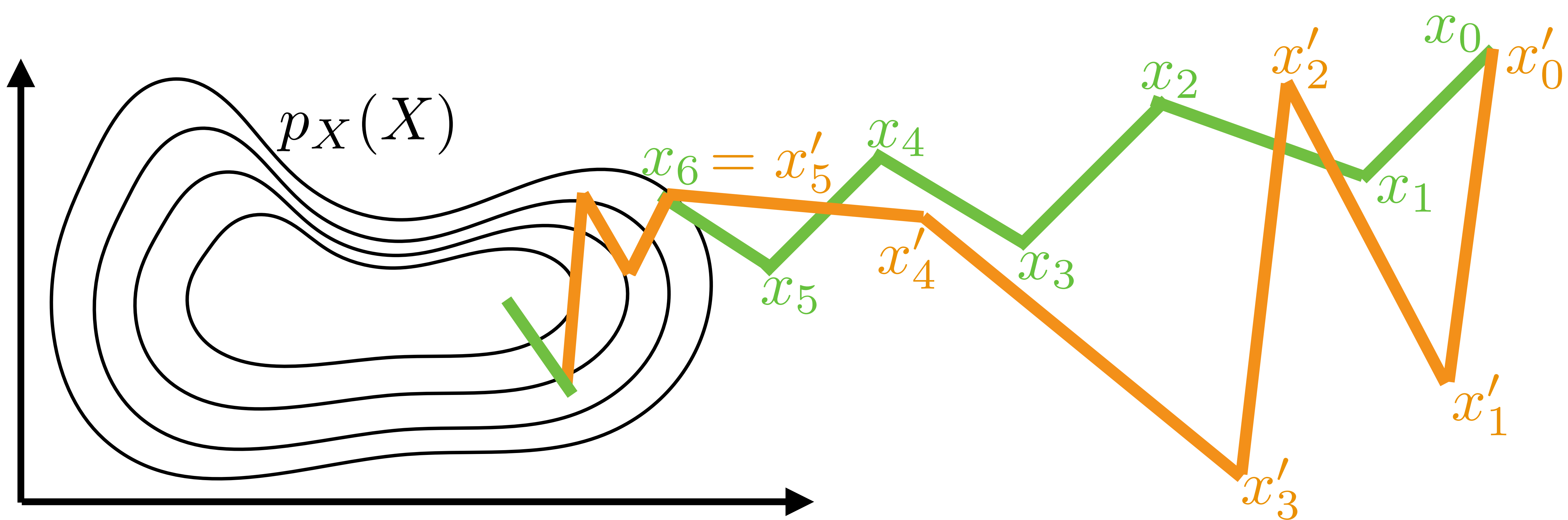
Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time
- Coupling [Jacob et al 2020]: make two MCMC chains; set up joint dynamics so chains meet exactly & quickly



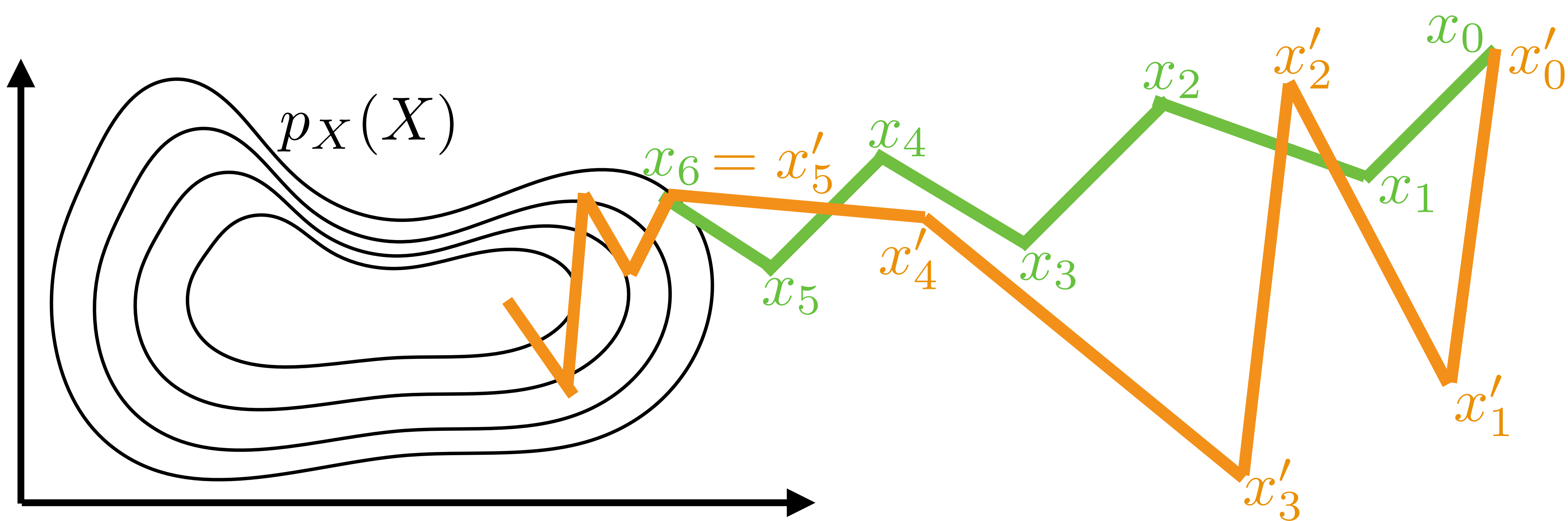
Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time
- Coupling [Jacob et al 2020]: make two MCMC chains; set up joint dynamics so chains meet exactly & quickly



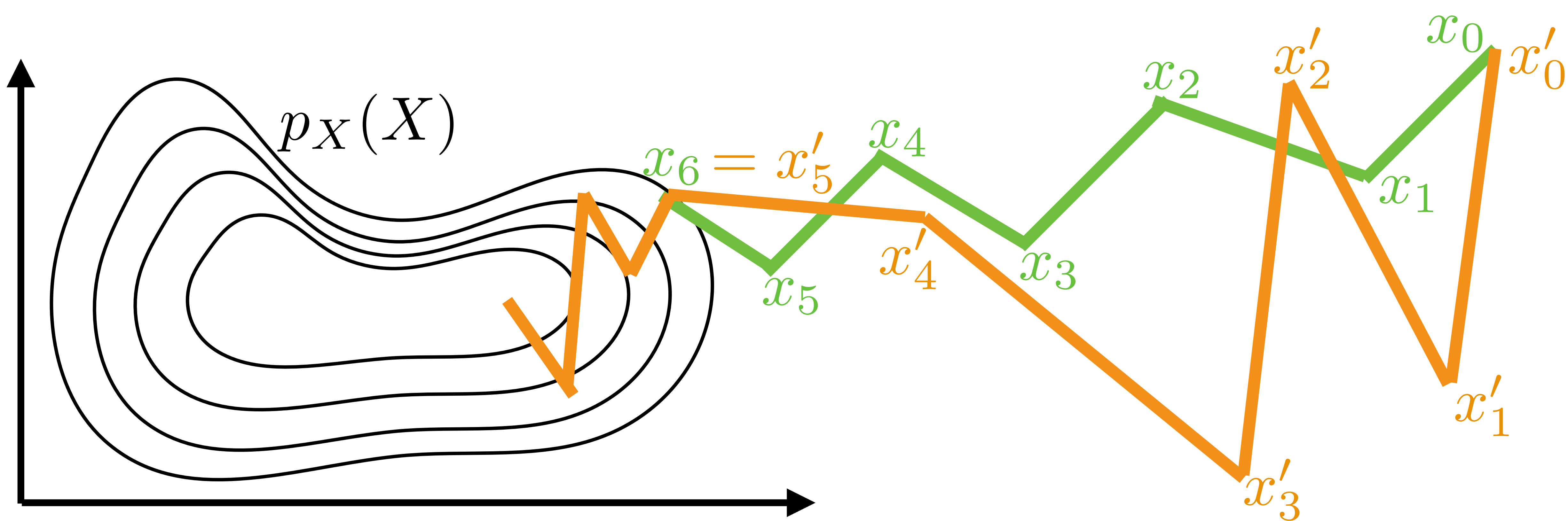
Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time
- Coupling [Jacob et al 2020]: make two MCMC chains; set up joint dynamics so chains meet exactly & quickly



Coupling for removing bias

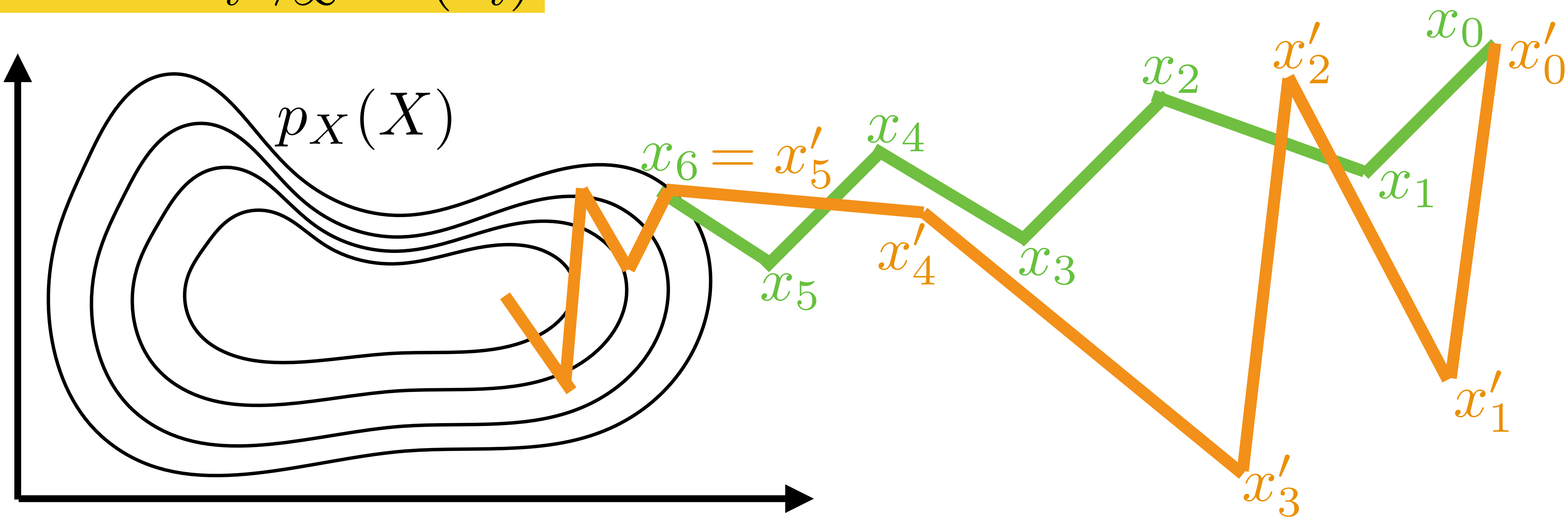
- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time
- Coupling [Jacob et al 2020]: make two MCMC chains; set up joint dynamics so chains meet exactly & quickly; then subtract out the bias of one using the other [Glynn, Rhee 2014]



Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T}\sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time
- Coupling [Jacob et al 2020]: make two MCMC chains; set up joint dynamics so chains meet exactly & quickly; then subtract out the bias of one using the other [Glynn, Rhee 2014]

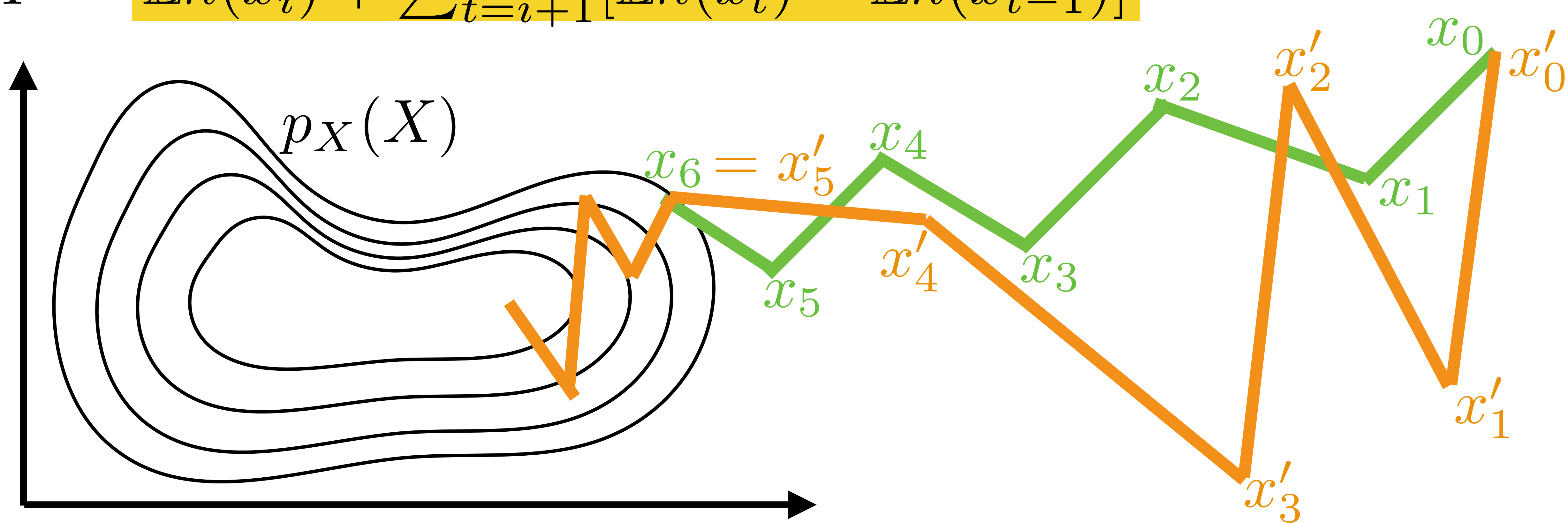
$H^* = \lim_{t\to\infty} \mathbb{E} h(x_t)$



Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time
- Coupling [Jacob et al 2020]: make two MCMC chains; set up joint dynamics so chains meet exactly & quickly; then subtract out the bias of one using the other [Glynn, Rhee 2014]

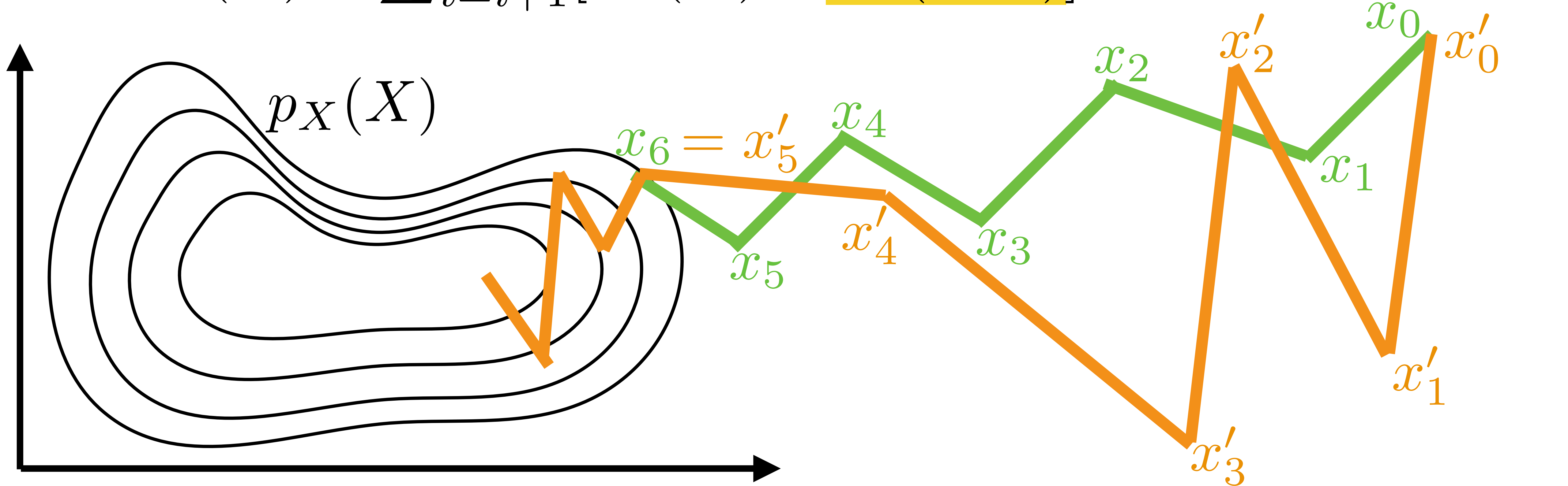
$$H^* = \mathbb{E}h(x_i) + \sum_{t=i+1}^{\infty} [\mathbb{E}h(x_t) - \mathbb{E}h(x_{t-1})]$$



Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time
- Coupling [Jacob et al 2020]: make two MCMC chains; set up joint dynamics so chains meet exactly & quickly; then subtract out the bias of one using the other [Glynn, Rhee 2014]

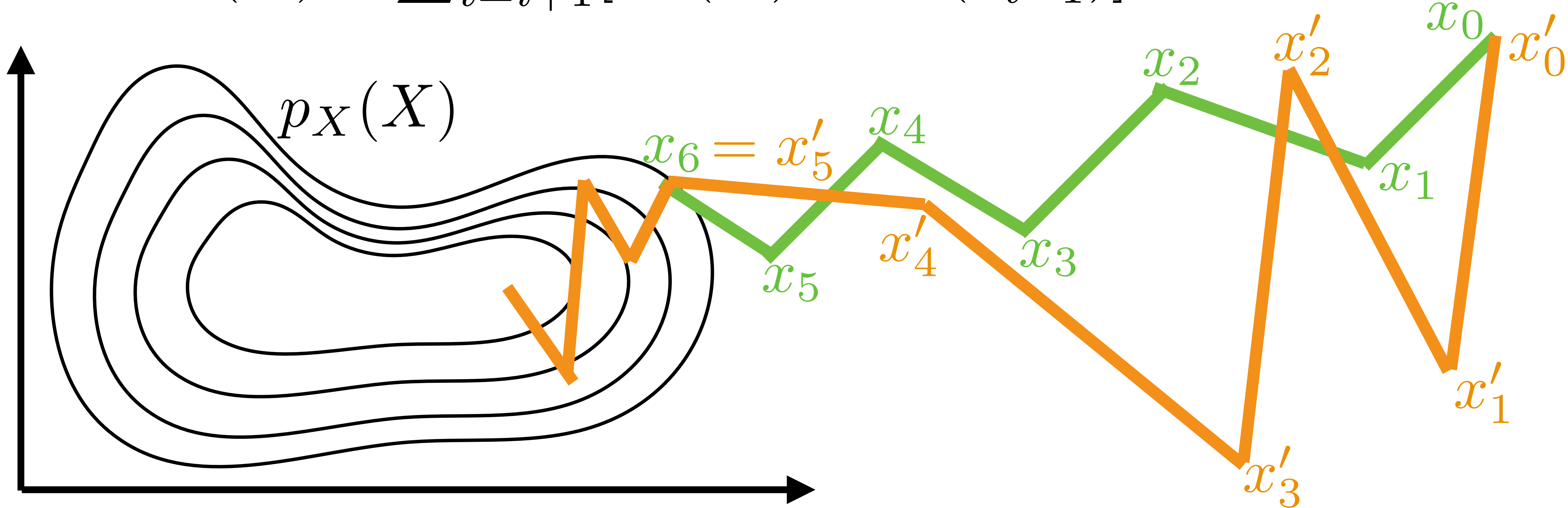
$H^* = \mathbb{E}h(x_i) + \sum_{t=i+1}^{\infty} [\mathbb{E}h(x_t) - \mathbb{E}h(x_{t-1})]$



Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time
- Coupling [Jacob et al 2020]: make two MCMC chains; set up joint dynamics so chains meet exactly & quickly; then subtract out the bias of one using the other [Glynn, Rhee 2014]

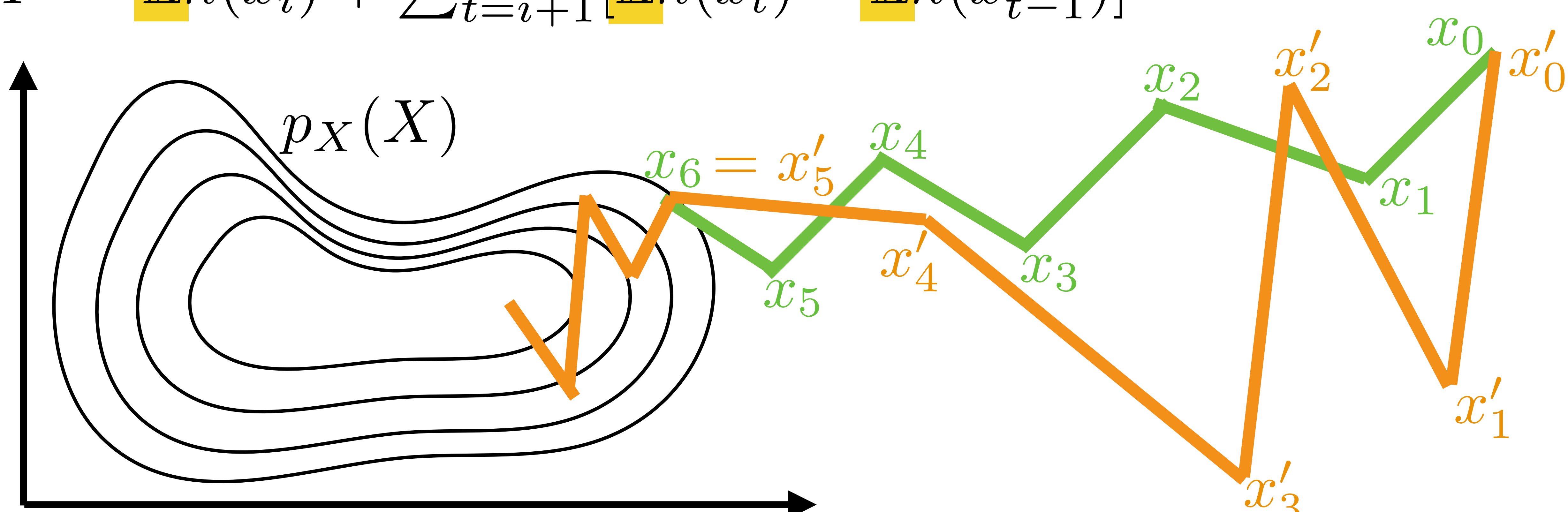
$$H^* = \mathbb{E} h(x_i) + \sum_{t=i+1}^{\infty} [\mathbb{E} h(x_t) - \mathbb{E} h(x'_{t-1})]$$



Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T}\sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time
- Coupling [Jacob et al 2020]: make two MCMC chains; set up joint dynamics so chains meet exactly & quickly; then subtract out the bias of one using the other [Glynn, Rhee 2014]

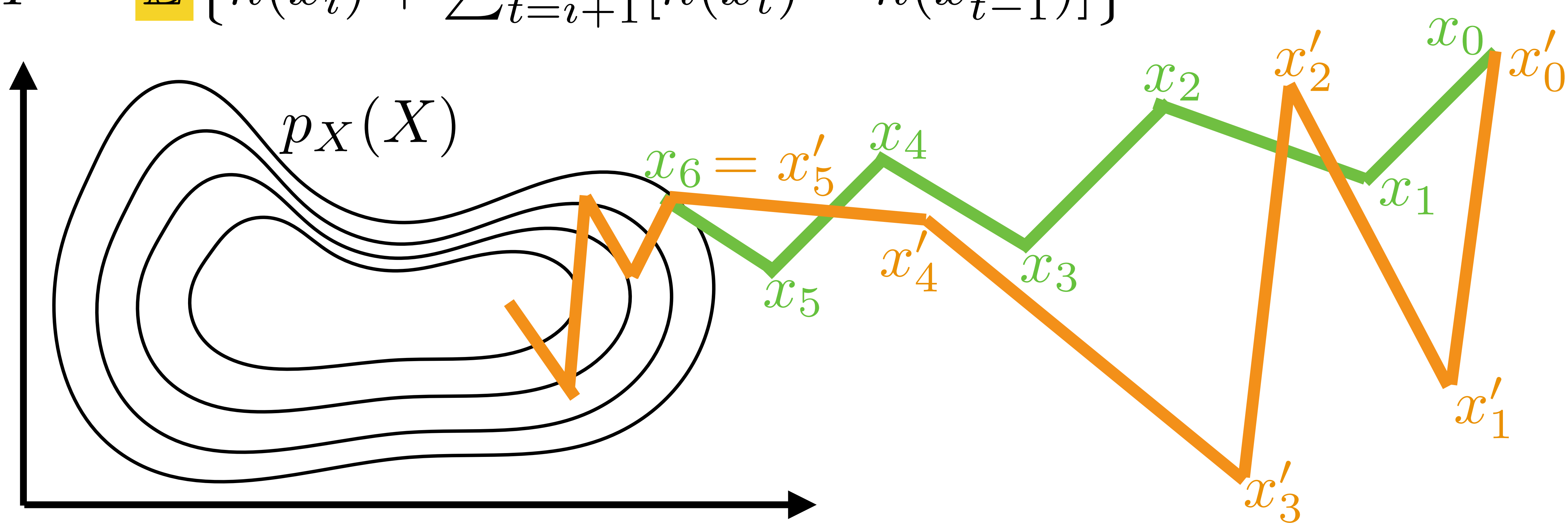
$$H^* = \mathbb{E}h(x_i) + \sum_{t=i+1}^{\infty} [\mathbb{E}h(x_t) - \mathbb{E}h(x'_{t-1})]$$



Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time
- Coupling [Jacob et al 2020]: make two MCMC chains; set up joint dynamics so chains meet exactly & quickly; then subtract out the bias of one using the other [Glynn, Rhee 2014]

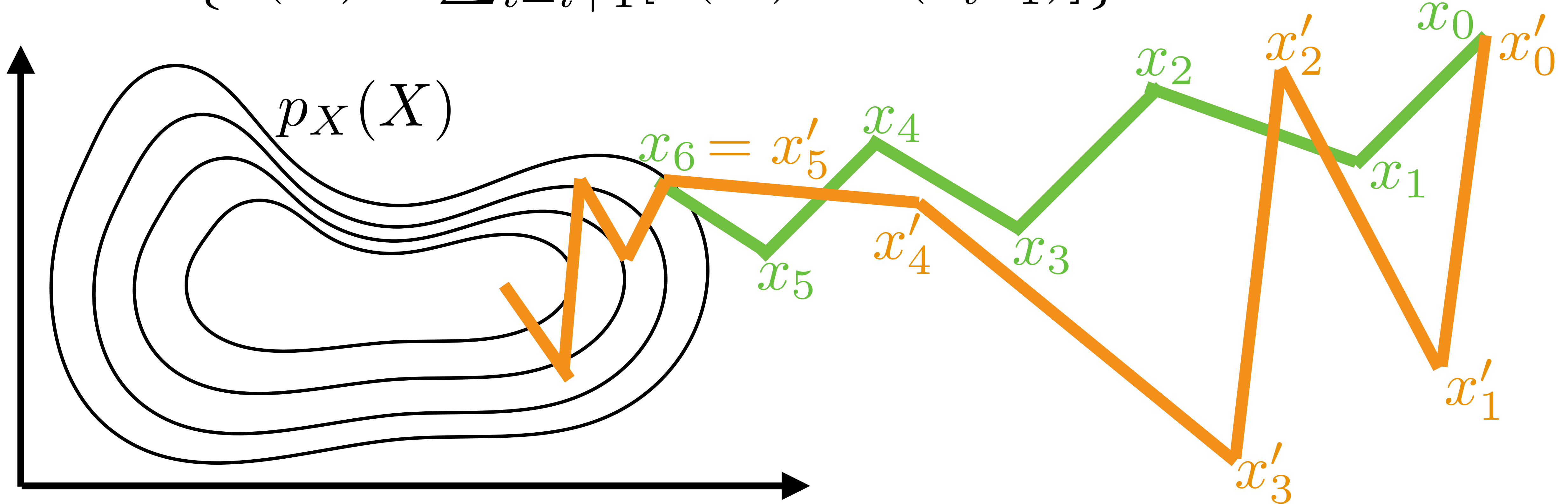
$$H^* = \mathbb{E}\{h(x_i) + \sum_{t=i+1}^{\infty} [h(x_t) - h(x'_{t-1})]\}$$



Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time
- Coupling [Jacob et al 2020]: make two MCMC chains; set up joint dynamics so chains meet exactly & quickly; then subtract out the bias of one using the other [Glynn, Rhee 2014]

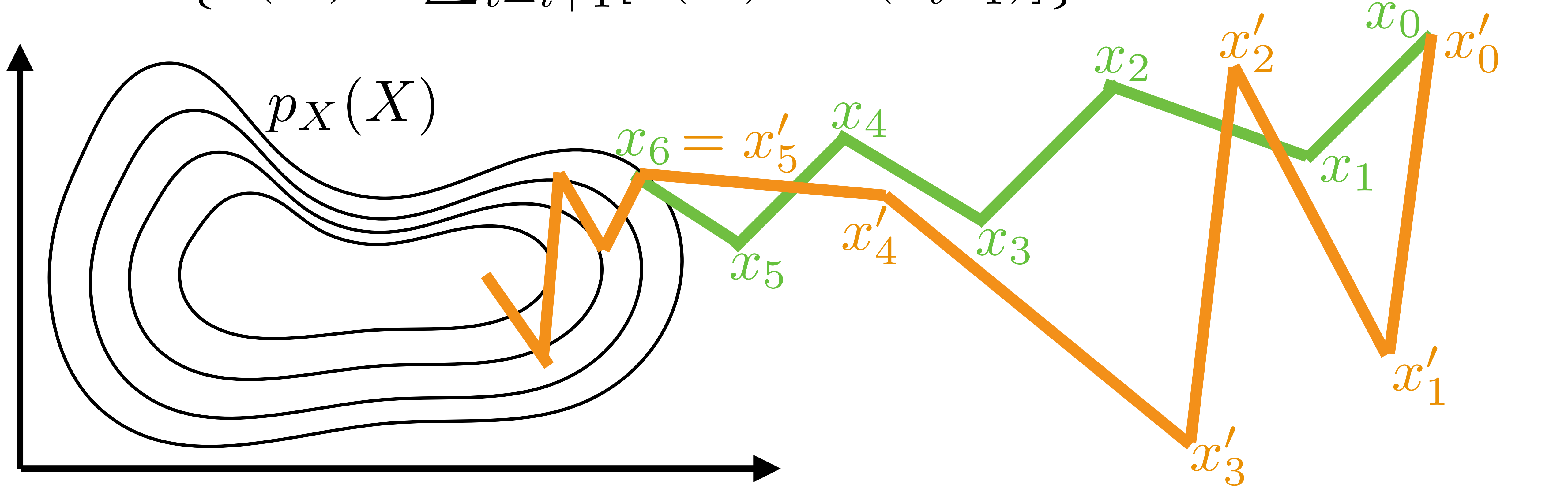
$$H^* = \mathbb{E}\left\{h(x_i) + \sum_{t=i+1}^{\infty} [h(x_t) - h(x'_{t-1})]\right\}$$



Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
    - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time
- Coupling [Jacob et al 2020]: make two MCMC chains; set up joint dynamics so chains meet exactly & quickly; then subtract out the bias of one using the other [Glynn, Rhee 2014]

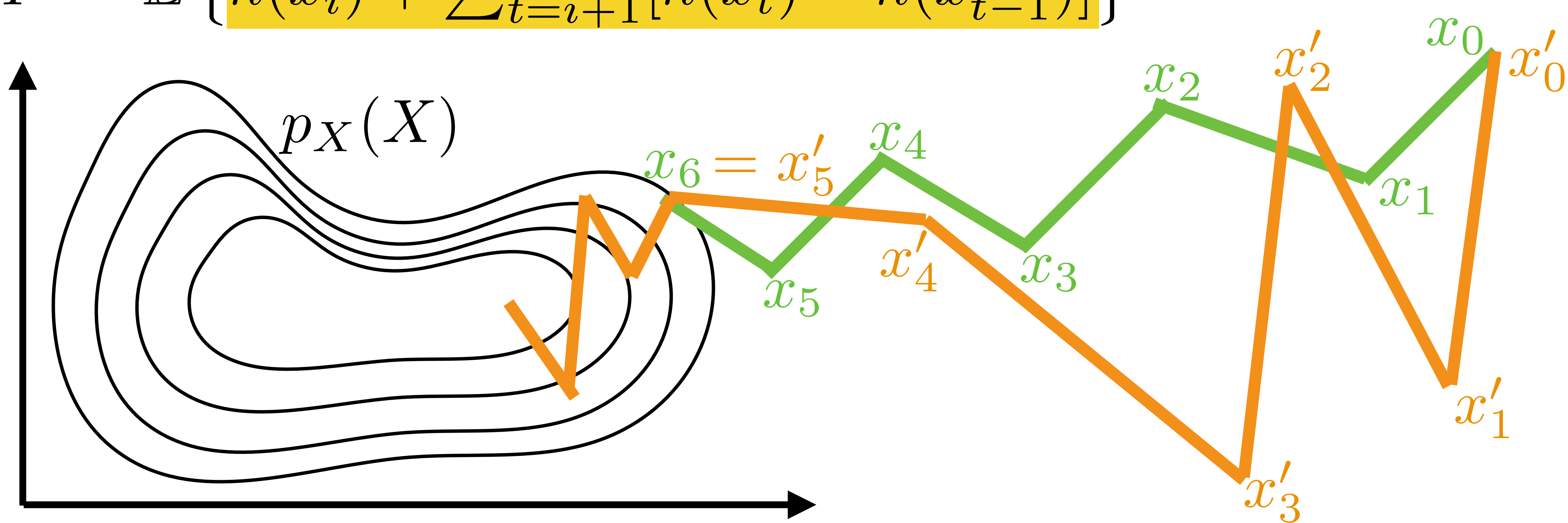
$$H^* = \mathbb{E}\left\{h(x_i) + \sum_{t=i+1}^{\tau} [h(x_t) - h(x'_{t-1})]\right\}$$



Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time
- Coupling [Jacob et al 2020]: make two MCMC chains; set up joint dynamics so chains meet exactly & quickly; then subtract out the bias of one using the other [Glynn, Rhee 2014]

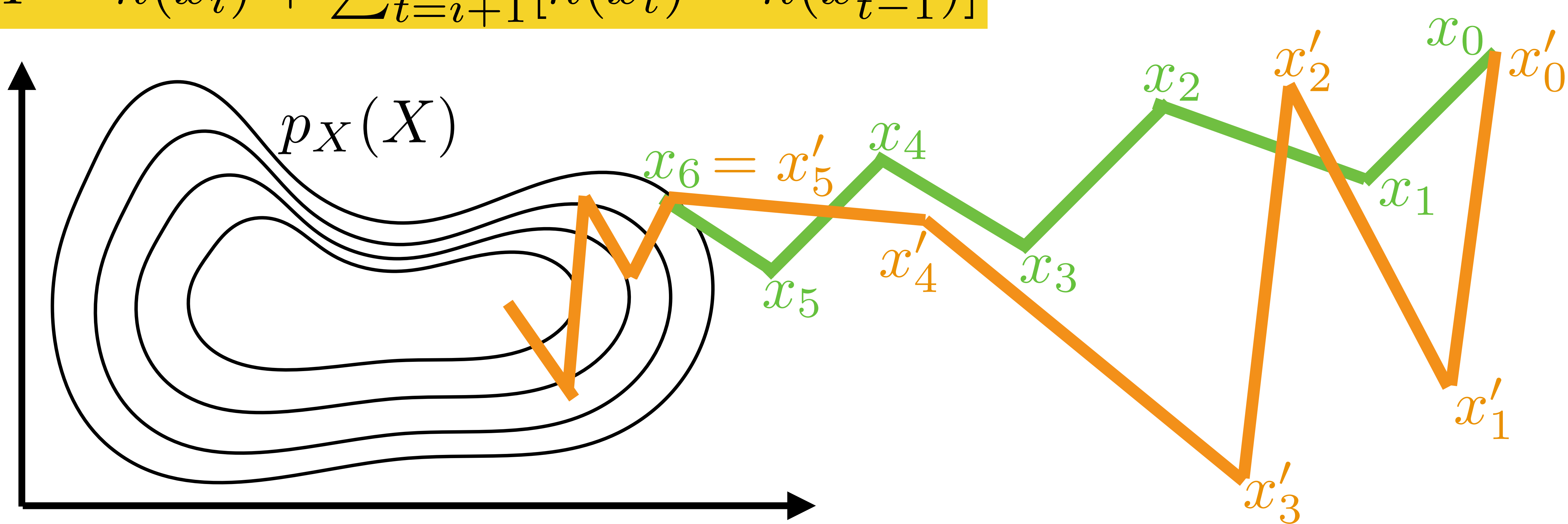
$$H^* = \mathbb{E}\left\{h(x_i) + \sum_{t=i+1}^{\tau}[h(x_t) - h(x'_{t-1})]\right\}$$



Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time
- Coupling [Jacob et al 2020]: make two MCMC chains; set up joint dynamics so chains meet exactly & quickly; then subtract out the bias of one using the other [Glynn, Rhee 2014]

$$\hat{H} = h(x_i) + \sum_{t=i+1}^{\tau} [h(x_t) - h(x'_{t-1})]$$

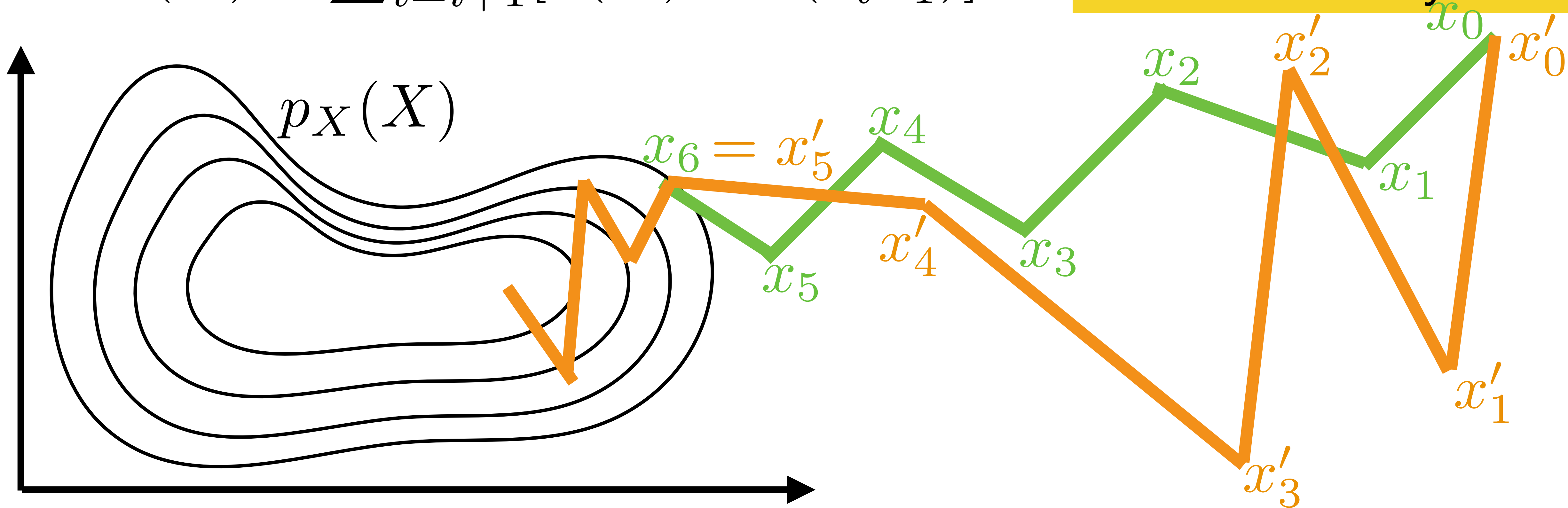


Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible $\longrightarrow \hat{H} = \frac{1}{T}\sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time
- Coupling [Jacob et al 2020]: make two MCMC chains; set up joint dynamics so chains meet exactly & quickly; then subtract out the bias of one using the other [Glynn, Rhee 2014]

$$\hat{H} = h(x_i) + \sum_{t=i+1}^{\tau}[h(x_t) - h(x'_{t-1})]$$

How can they meet?

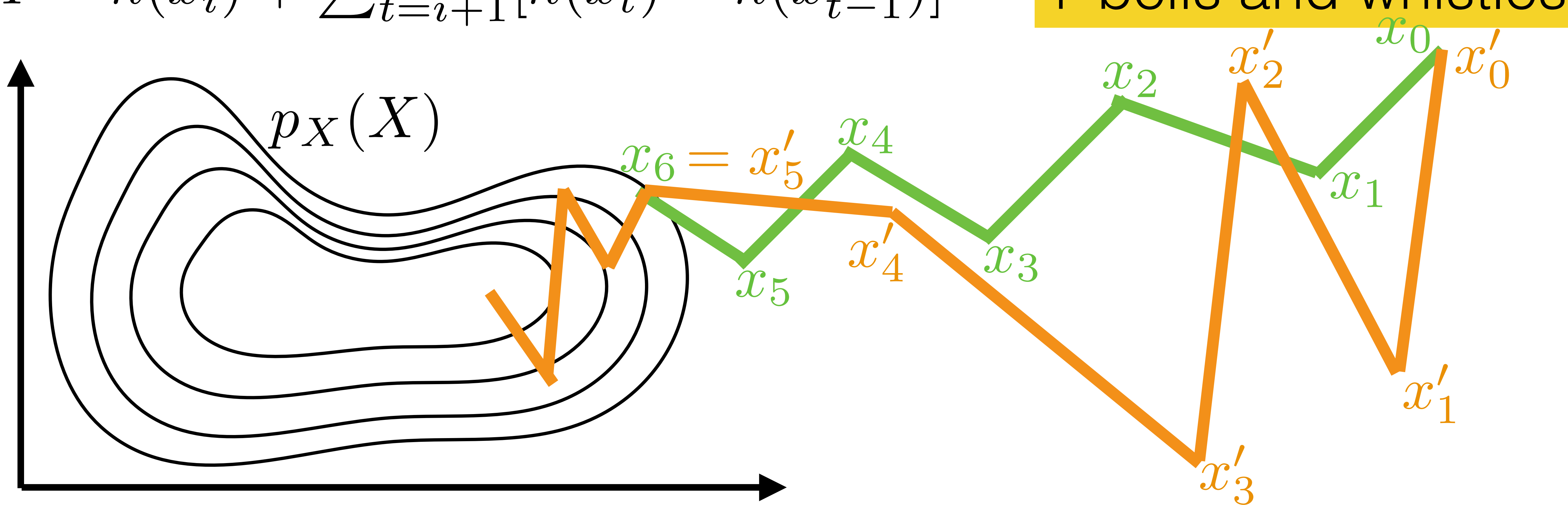


Coupling for removing bias

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Monte Carlo: unbiased but infeasible → $\hat{H} = \frac{1}{T} \sum_{t=1}^{T} h(x_t)$
- Markov chain Monte Carlo: biased in finite time
- Coupling [Jacob et al 2020]: make two MCMC chains; set up joint dynamics so chains meet exactly & quickly; then subtract out the bias of one using the other [Glynn, Rhee 2014]

$\hat{H} = h(x_i) + \sum_{t=i+1}^{\tau} [h(x_t) - h(x'_{t-1})]$

+ bells and whistles



- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion

Coupling for partitions: the challenge

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion

Coupling for partitions: the challenge

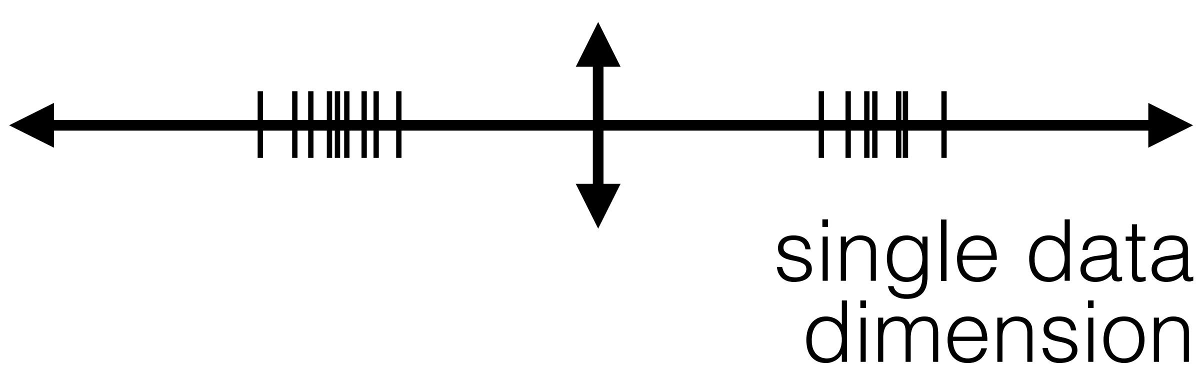
- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Coupling for partitions for unbiased MCMC approximation: not done in past work

Coupling for partitions: the challenge

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Coupling for partitions for unbiased MCMC approximation: not done in past work
- Couplings made for other purposes [Jerrum 1998, Gibbs 2004] could be used here, but they are on the label space → slow to meet

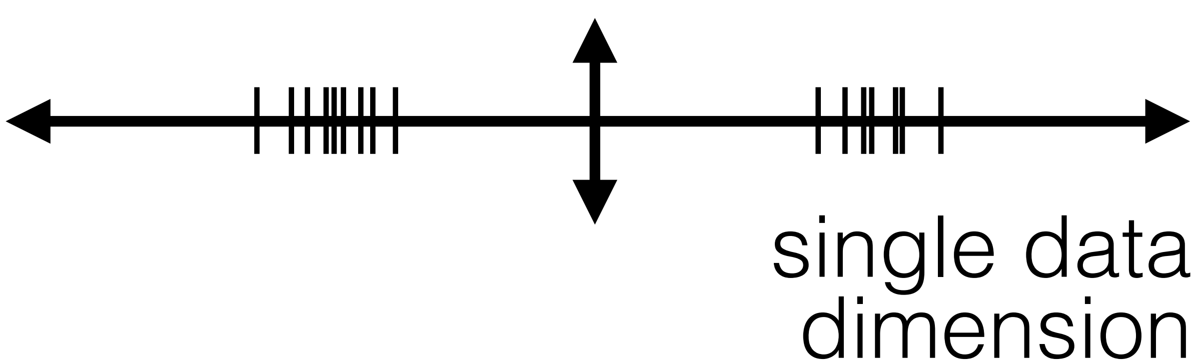
Coupling for partitions: the challenge

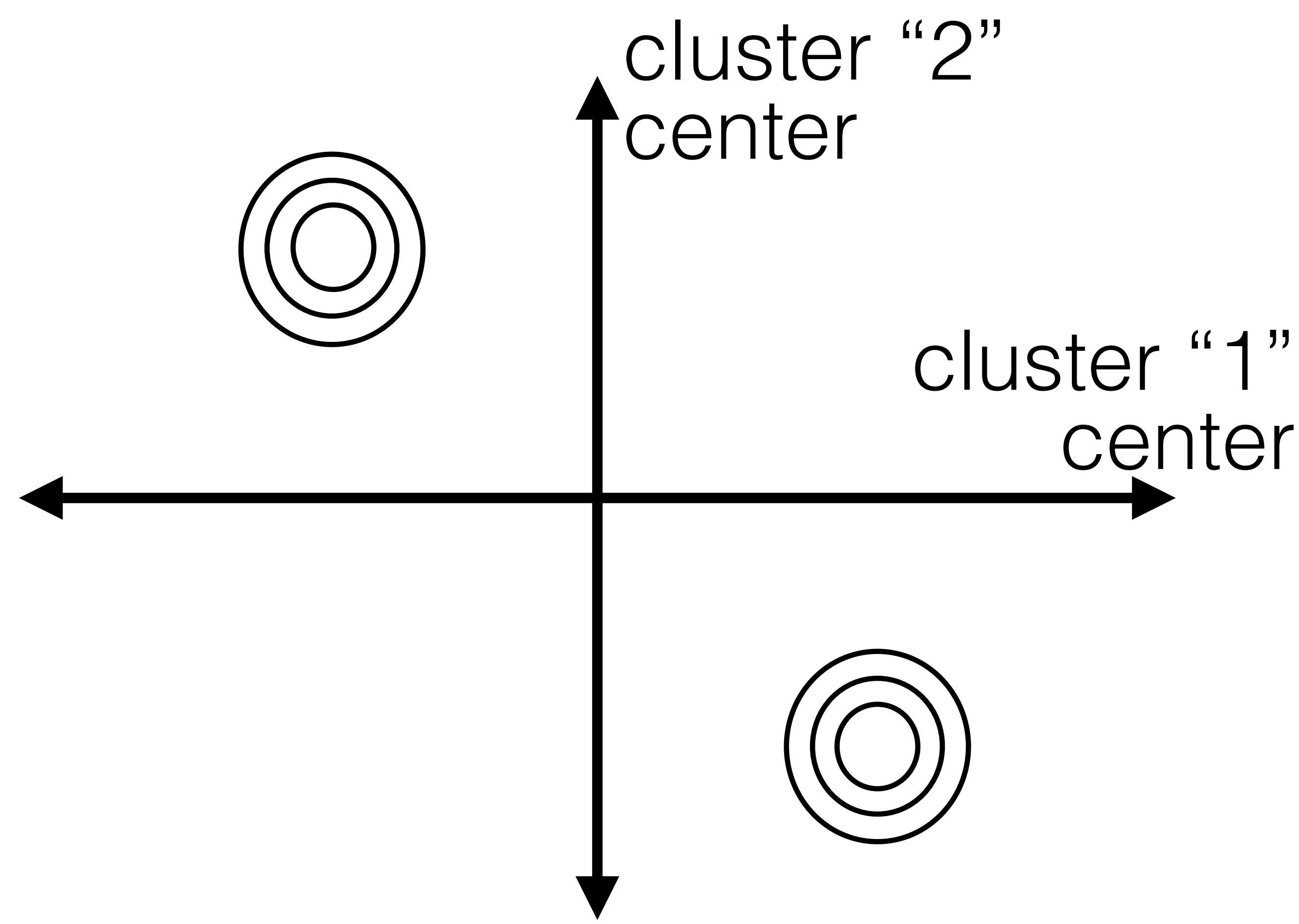
- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Coupling for partitions for unbiased MCMC approximation: not done in past work
- Couplings made for other purposes [Jerrum 1998, Gibbs 2004] could be used here, but they are on the label space → slow to meet



Coupling for partitions: the challenge

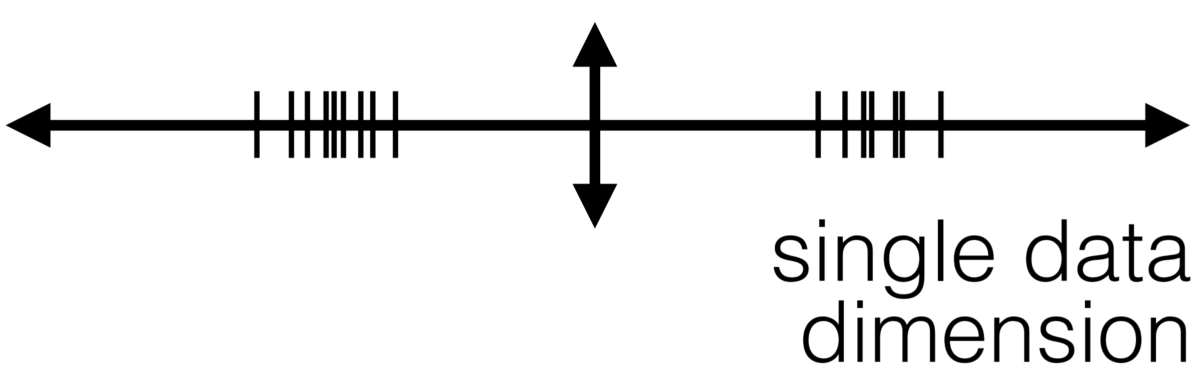
- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Coupling for partitions for unbiased MCMC approximation: not done in past work
- Couplings made for other purposes [Jerrum 1998, Gibbs 2004] could be used here, but they are on the label space → slow to meet

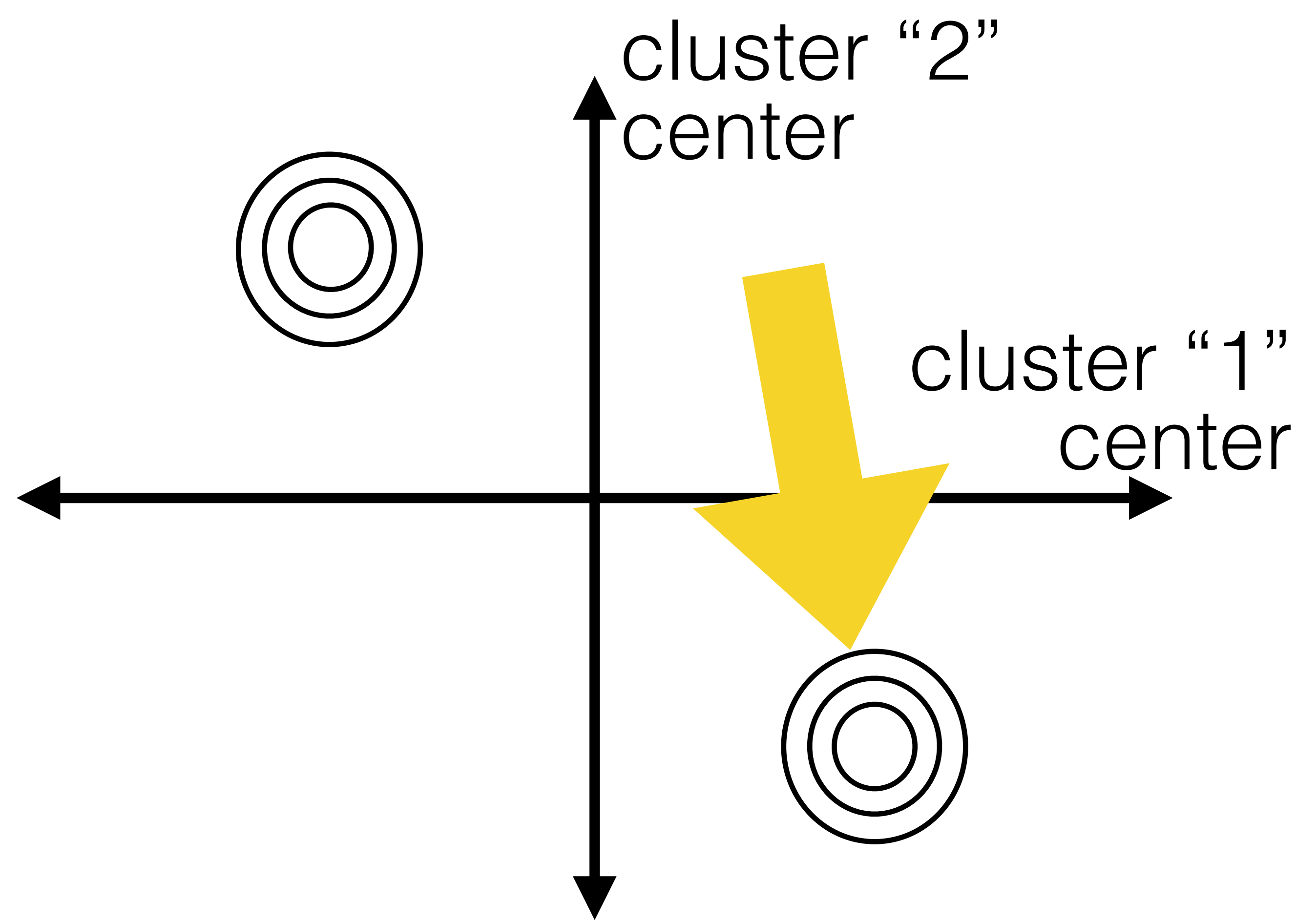




Coupling for partitions: the challenge

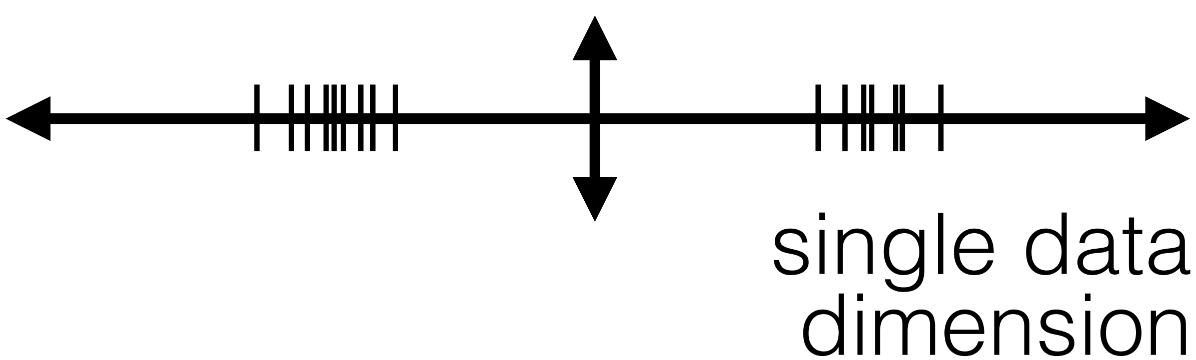
- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Coupling for partitions for unbiased MCMC approximation: not done in past work
- Couplings made for other purposes [Jerrum 1998, Gibbs 2004] could be used here, but they are on the label space → slow to meet

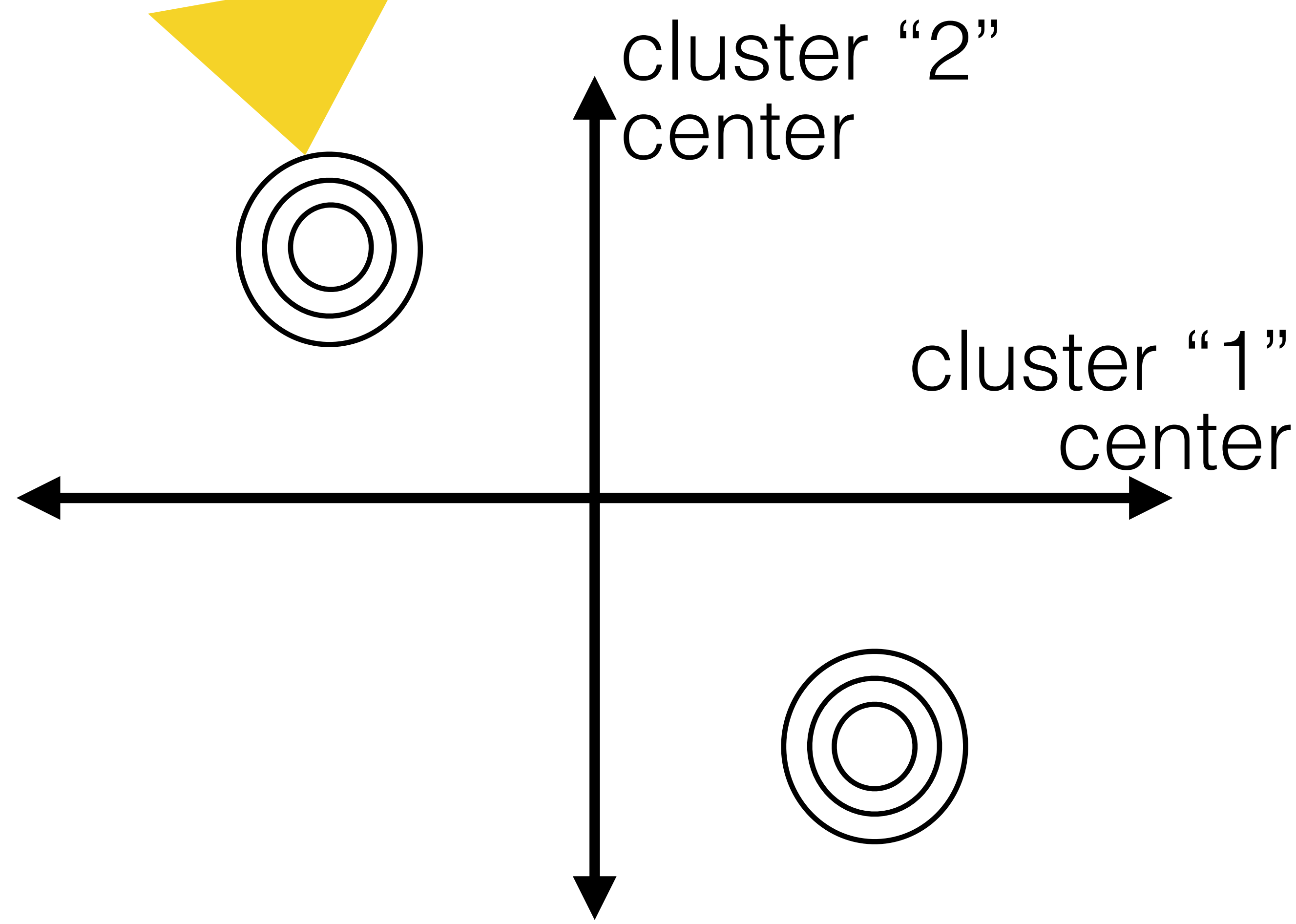




Coupling for partitions: the challenge

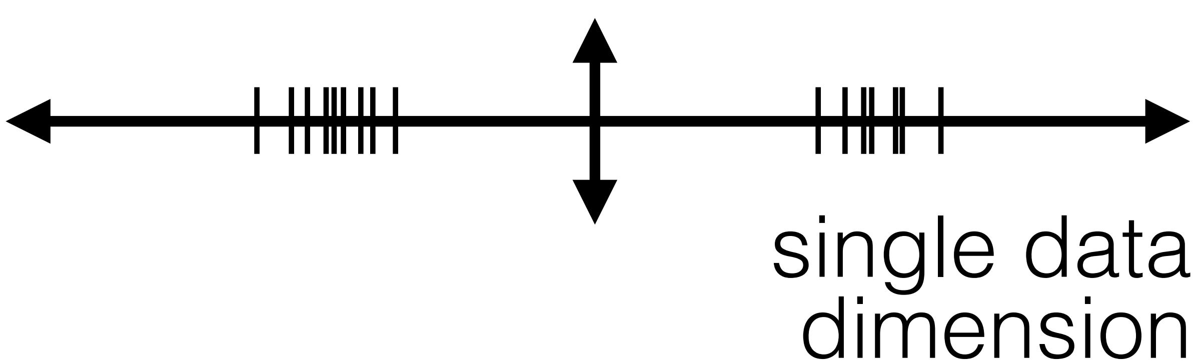
- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Coupling for partitions for unbiased MCMC approximation: not done in past work
- Couplings made for other purposes [Jerrum 1998, Gibbs 2004] could be used here, but they are on the label space → slow to meet

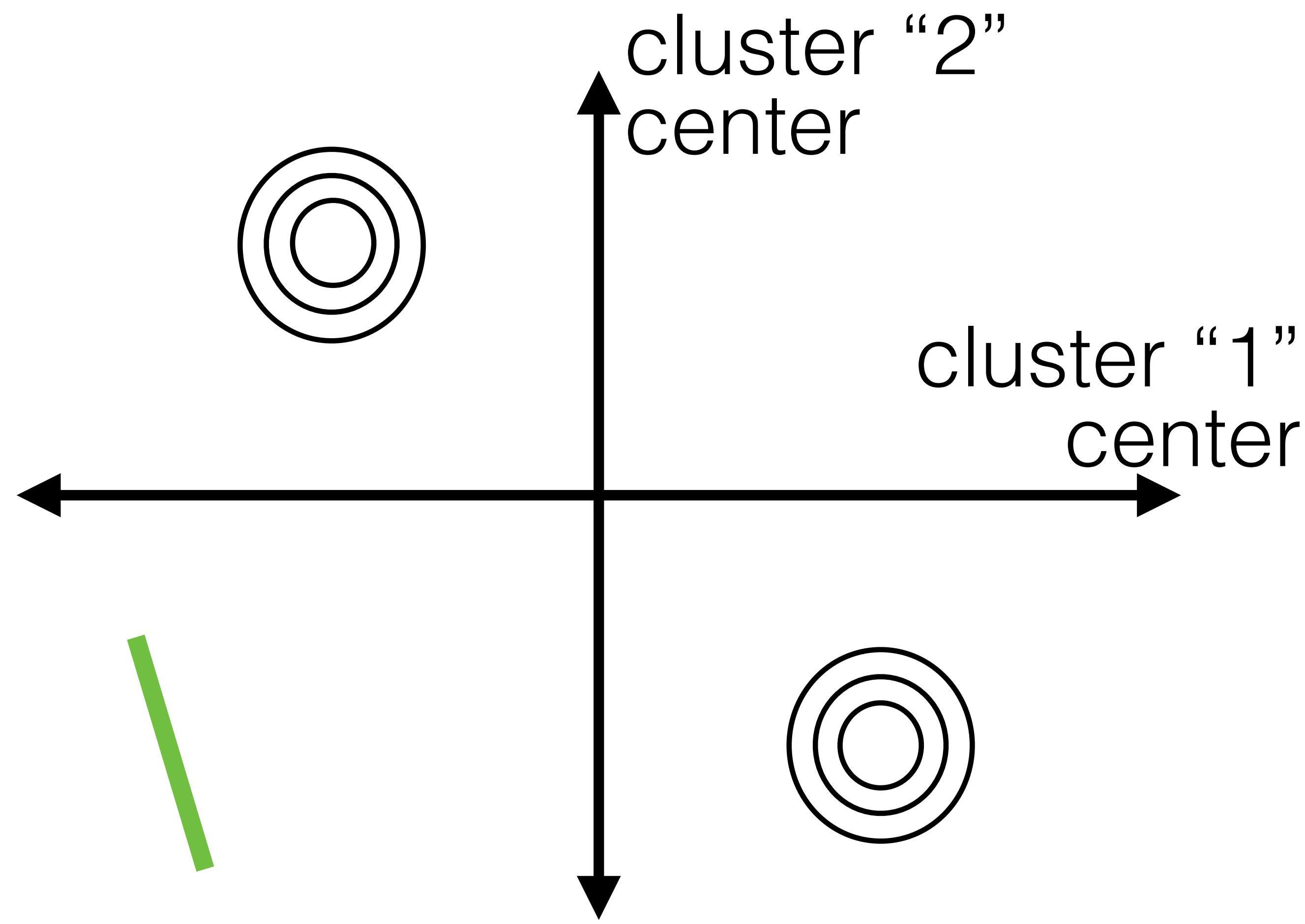




Coupling for partitions: the challenge

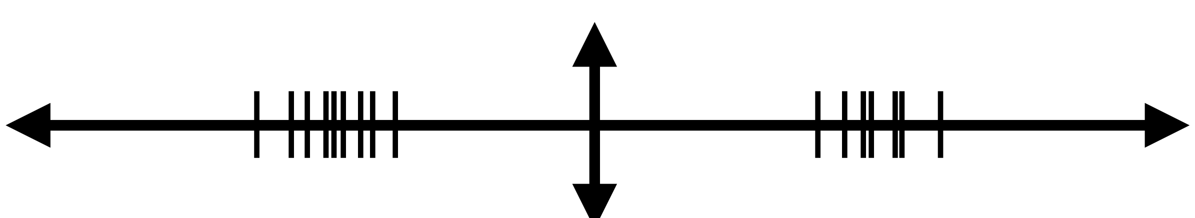
- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Coupling for partitions for unbiased MCMC approximation: not done in past work
- Couplings made for other purposes [Jerrum 1998, Gibbs 2004] could be used here, but they are on the label space → slow to meet

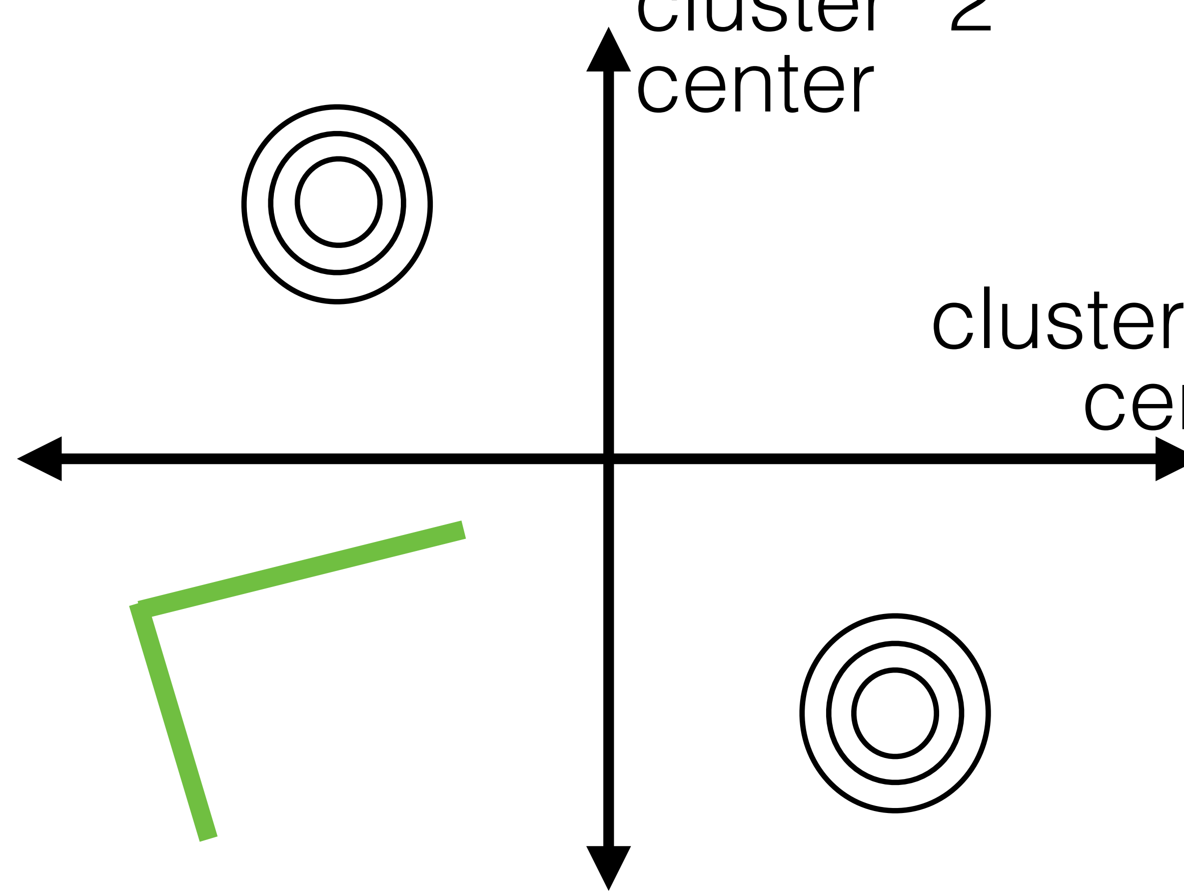




Coupling for partitions: the challenge

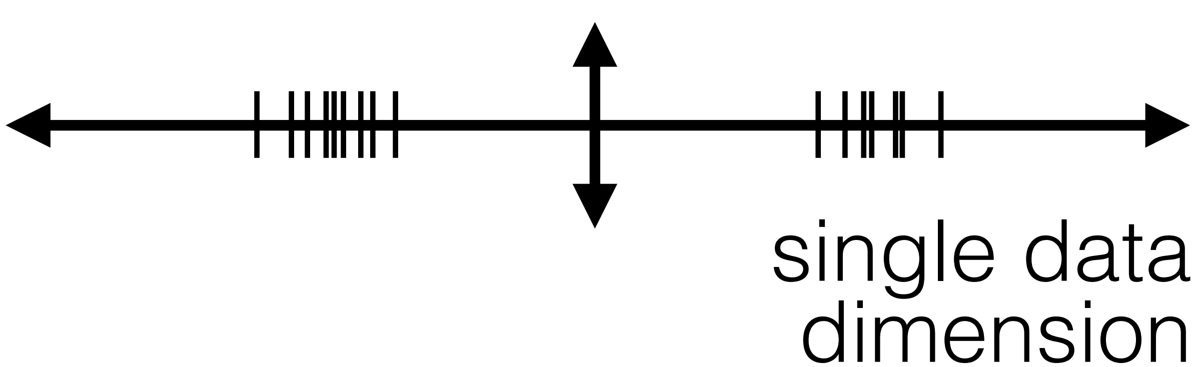
- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Coupling for partitions for unbiased MCMC approximation: not done in past work
- Couplings made for other purposes [Jerrum 1998, Gibbs 2004] could be used here, but they are on the label space → slow to meet

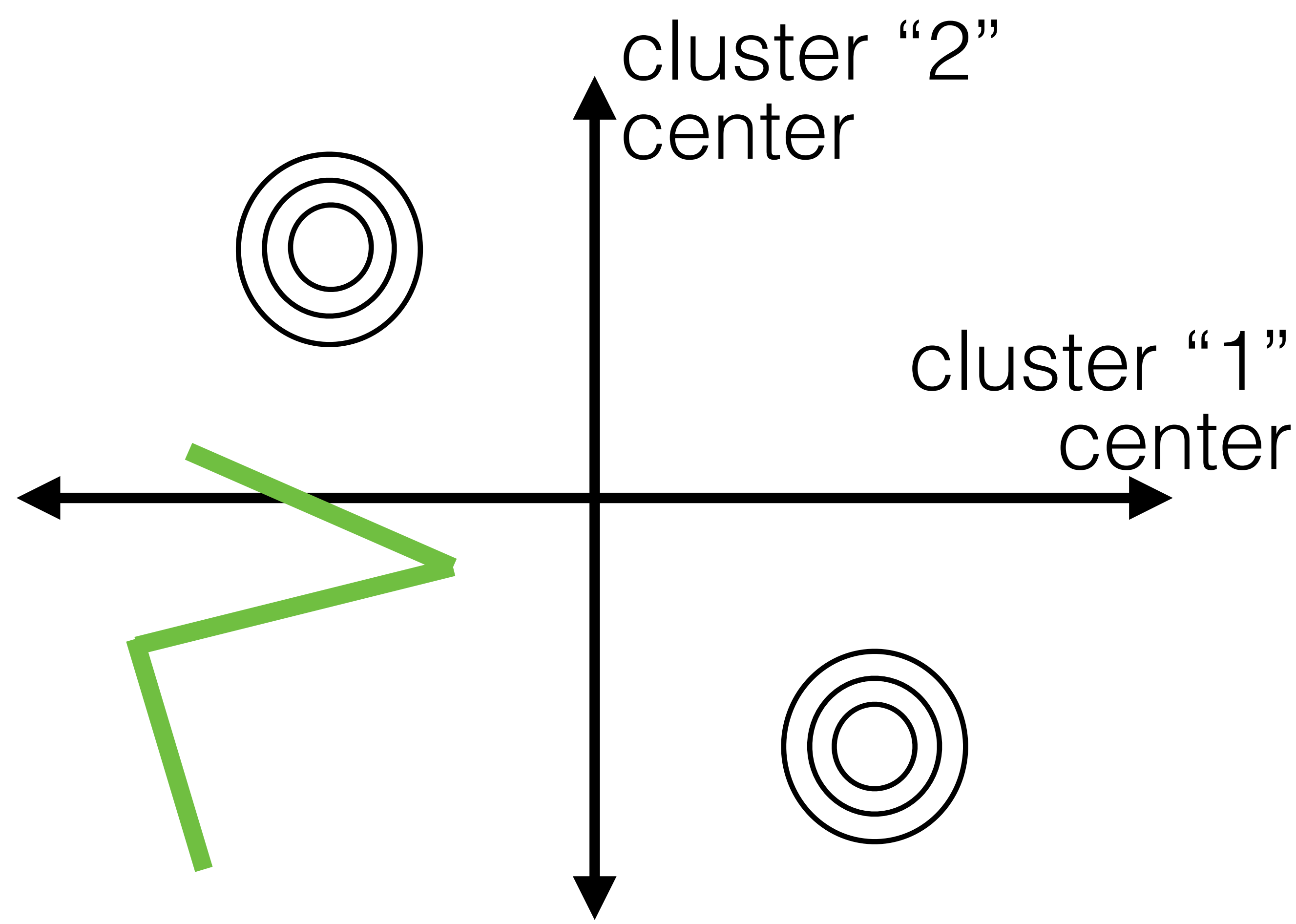




Coupling for partitions: the challenge

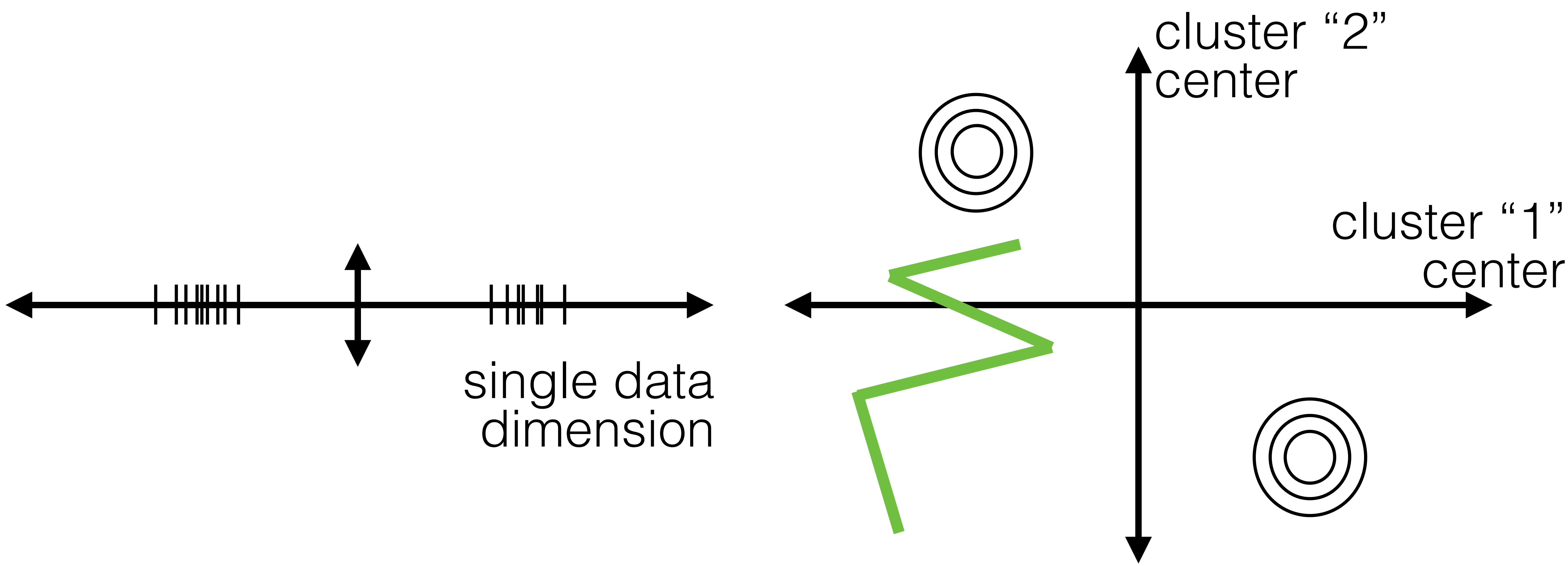
- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Coupling for partitions for unbiased MCMC approximation: not done in past work
- Couplings made for other purposes [Jerrum 1998, Gibbs 2004] could be used here, but they are on the label space → slow to meet





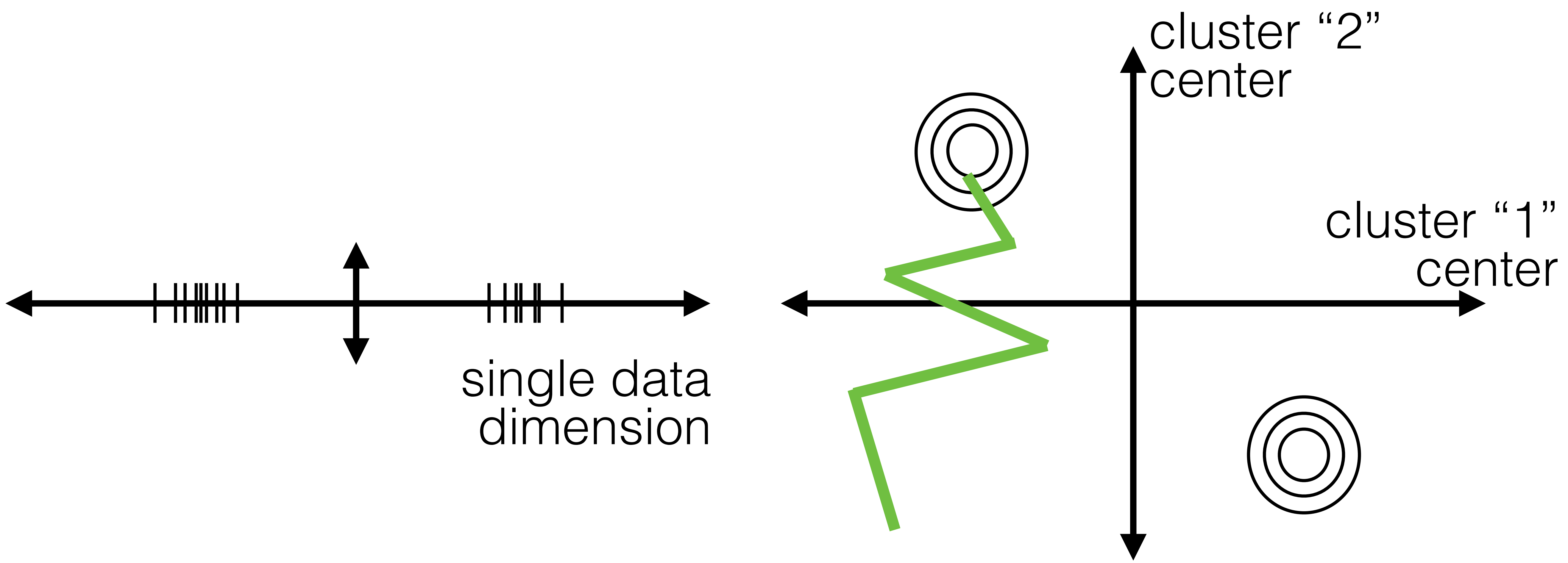
Coupling for partitions: the challenge

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Coupling for partitions for unbiased MCMC approximation: not done in past work
- Couplings made for other purposes [Jerrum 1998, Gibbs 2004] could be used here, but they are on the label space → slow to meet



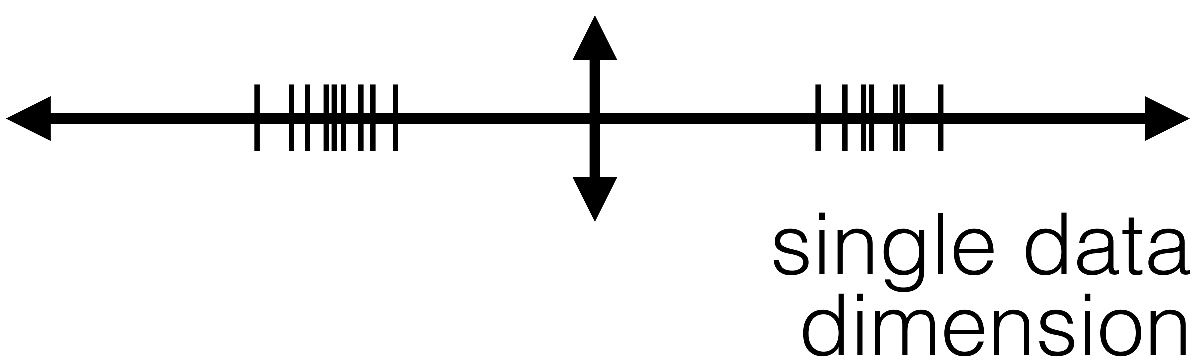
Coupling for partitions: the challenge

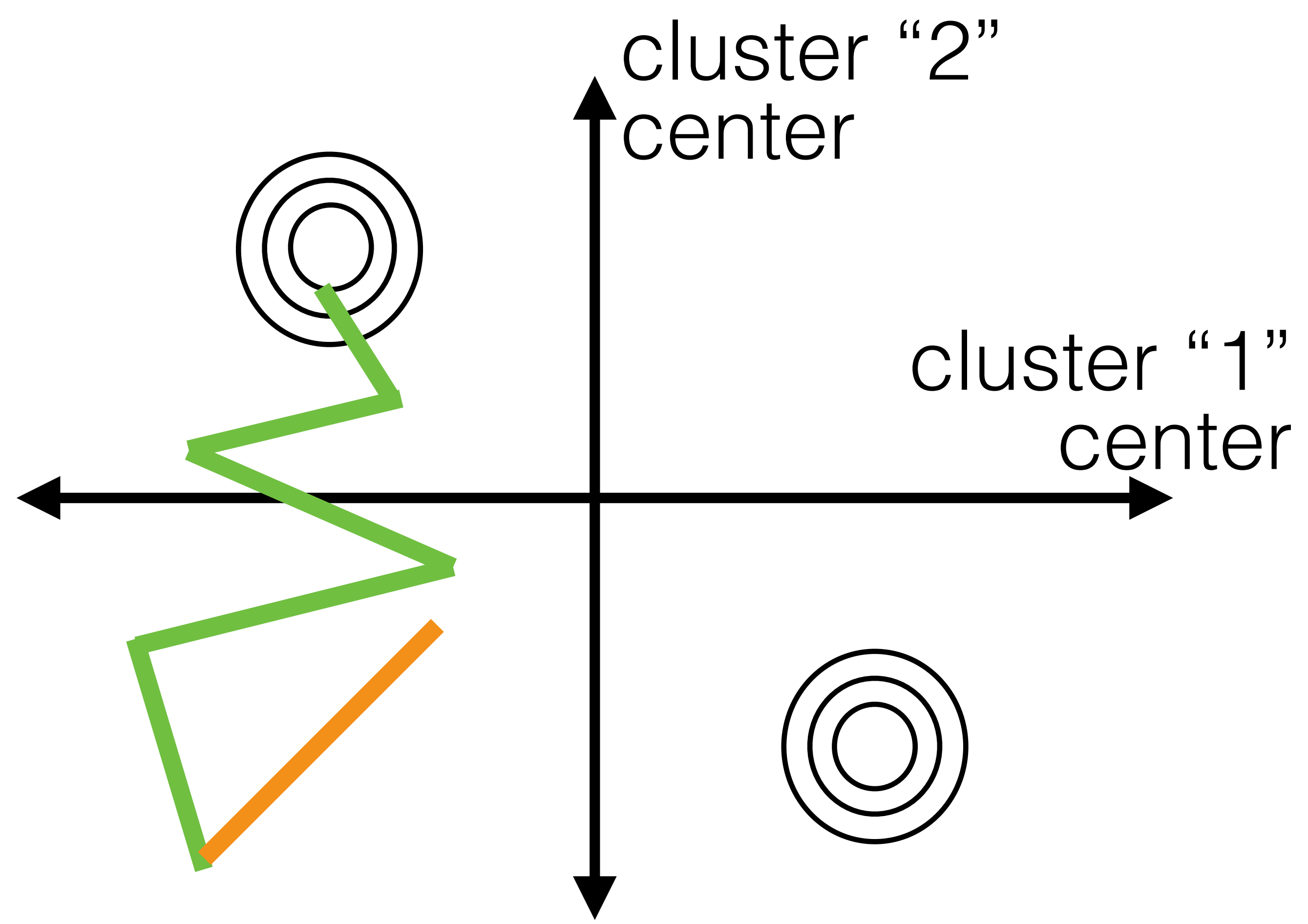
- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Coupling for partitions for unbiased MCMC approximation: not done in past work
- Couplings made for other purposes [Jerrum 1998, Gibbs 2004] could be used here, but they are on the label space → slow to meet



Coupling for partitions: the challenge

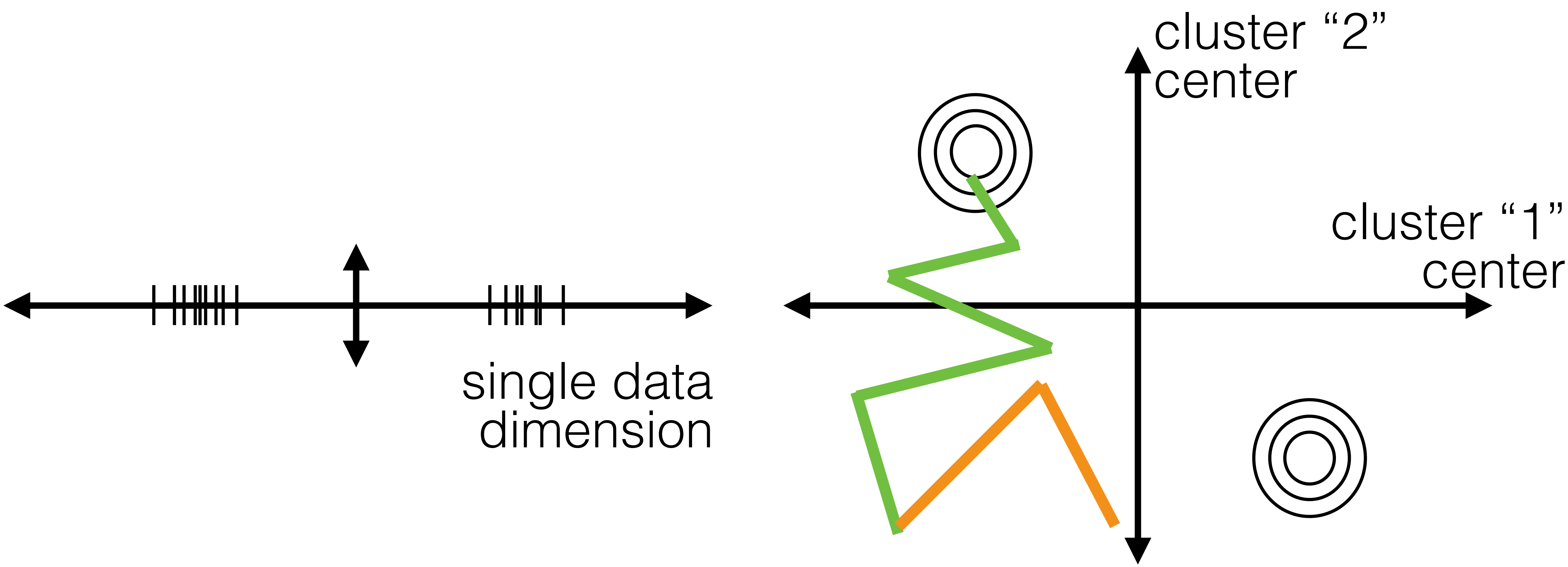
- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Coupling for partitions for unbiased MCMC approximation: not done in past work
- Couplings made for other purposes [Jerrum 1998, Gibbs 2004] could be used here, but they are on the label space → slow to meet





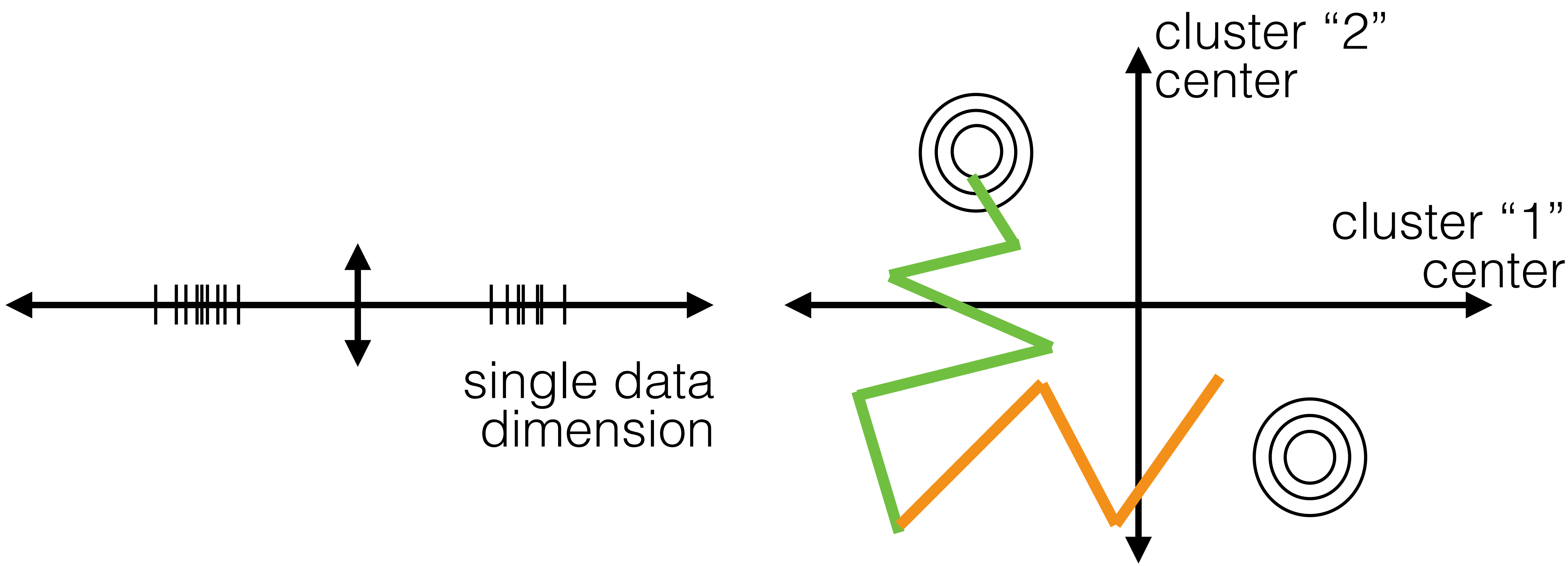
Coupling for partitions: the challenge

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Coupling for partitions for unbiased MCMC approximation: not done in past work
- Couplings made for other purposes [Jerrum 1998, Gibbs 2004] could be used here, but they are on the label space → slow to meet



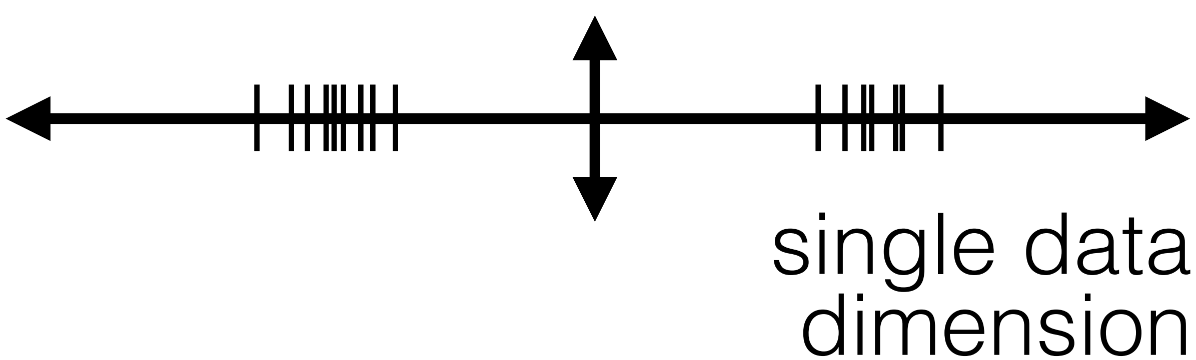
Coupling for partitions: the challenge

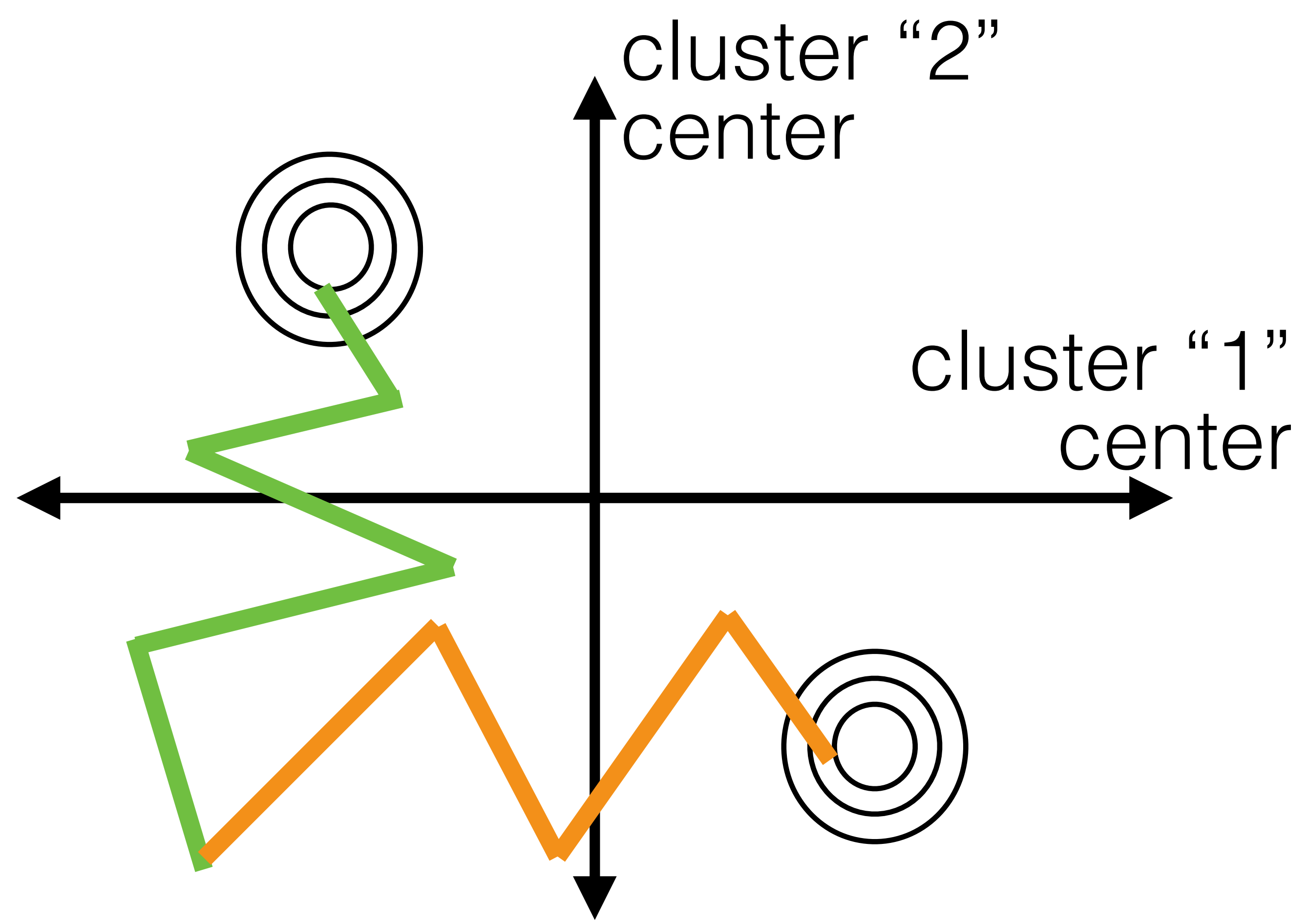
- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Coupling for partitions for unbiased MCMC approximation: not done in past work
- Couplings made for other purposes [Jerrum 1998, Gibbs 2004] could be used here, but they are on the label space → slow to meet



Coupling for partitions: the challenge

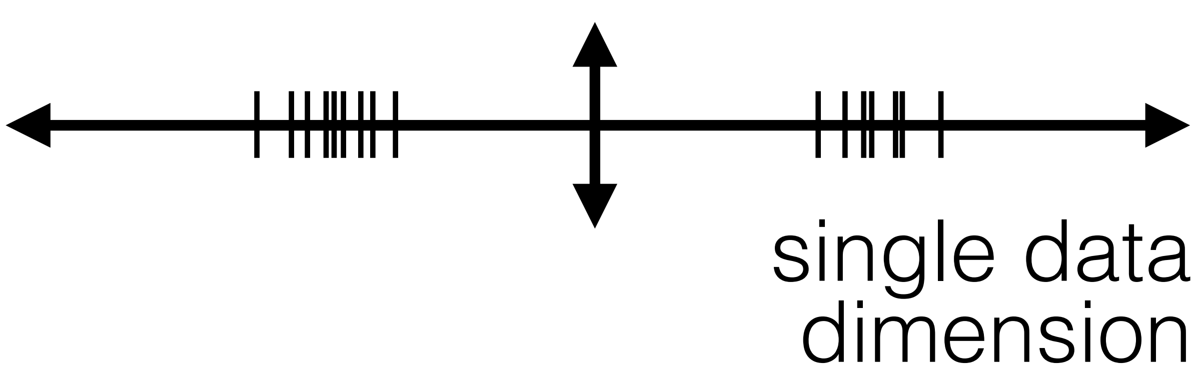
- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Coupling for partitions for unbiased MCMC approximation: not done in past work
- Couplings made for other purposes [Jerrum 1998, Gibbs 2004] could be used here, but they are on the label space → slow to meet

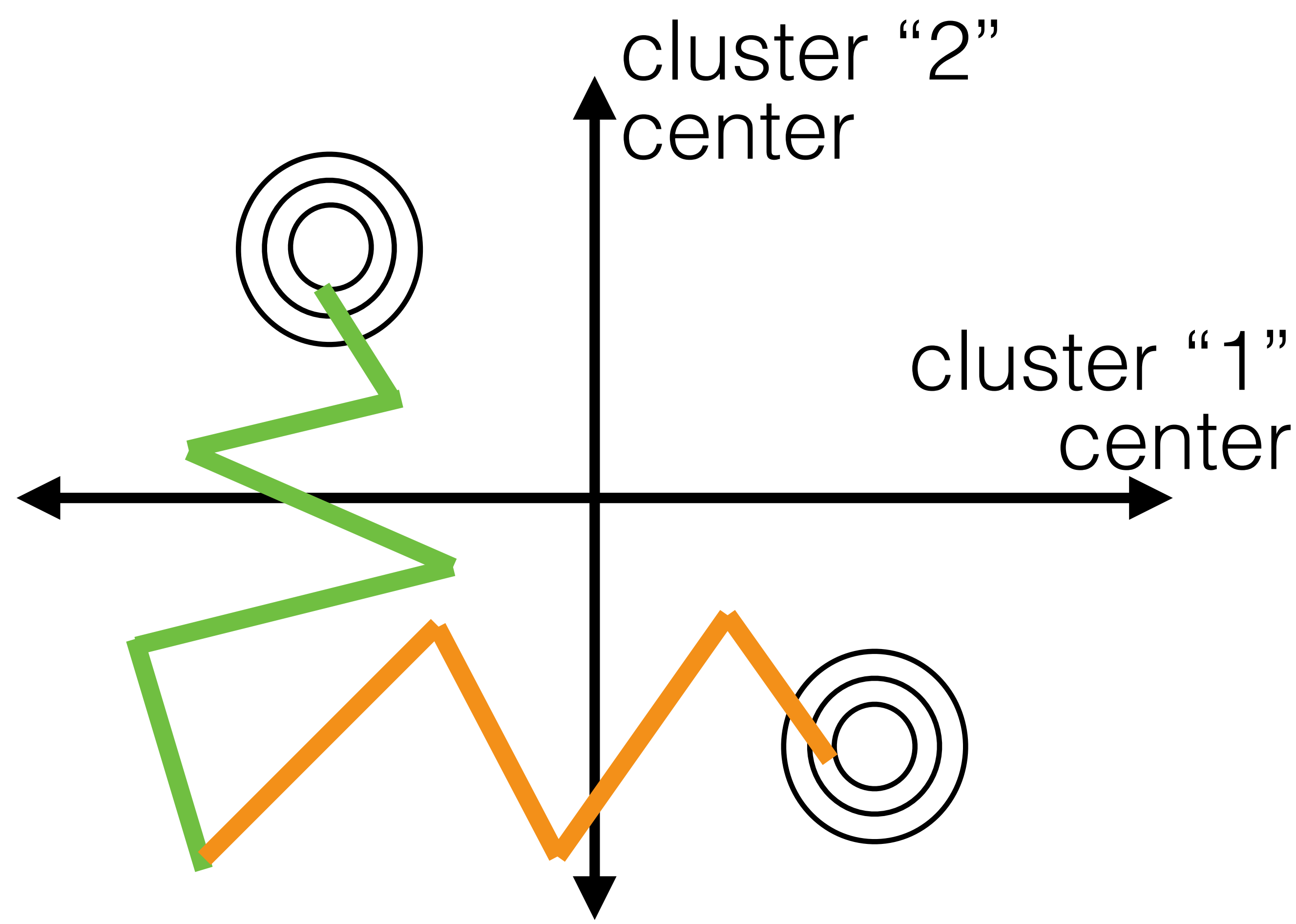




Coupling for partitions: the challenge

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- Coupling for partitions for unbiased MCMC approximation: not done in past work
- Couplings made for other purposes [Jerrum 1998, Gibbs 2004] could be used here, but they are on the label space → slow to meet
- Note: to switch labels, all the cluster assignments have to flip too





- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
    - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion

Coupling for partitions: our proposal

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion

Coupling for partitions: our proposal

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- We propose to work directly in the partition space

Coupling for partitions: our proposal

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- We propose to work directly in the partition space
- Given current partition, next is distributed:
  - 1st chain: $\sum_{k=1}^{K} a_k \delta_{\pi_k}$

Coupling for partitions: our proposal

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- We propose to work directly in the partition space
- Given current partition, next is distributed:
  - 1st chain: $\sum_{k=1}^{K} a_k \delta_{\pi_k}$
  - 2nd chain: $\sum_{k'=1}^{K'} b_{k'} \delta_{\nu_{k'}}$

Coupling for partitions: our proposal

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- We propose to work directly in the partition space
- Given current partition, next is distributed:
  - 1st chain: $\sum_{k=1}^{K} a_k \delta_{\pi_k}$
  - 2nd chain: $\sum_{k'=1}^{K'} b_{k'} \delta_{\nu_{k'}}$
- Let coupling $\gamma = \sum_k \sum_{k'} u_{k,k'} \delta_{(\pi_k, \nu_{k'})}$ have marginals:

$$\sum_k u_{k,k'} = b_{k'} \quad \& \quad \sum_{k'} u_{k,k'} = a_k$$

Coupling for partitions: our proposal

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- We propose to work directly in the partition space
- Given current partition, next is distributed:
  - 1st chain: $\sum_{k=1}^{K} a_k \delta_{\pi_k}$
  - 2nd chain: $\sum_{k'=1}^{K'} b_{k'} \delta_{\nu_{k'}}$
- Let coupling $\gamma = \sum_k \sum_{k'} u_{k,k'} \delta_{(\pi_k, \nu_{k'})}$ have marginals:

$$\sum_k u_{k,k'} = b_{k'} \quad \& \quad \sum_{k'} u_{k,k'} = a_k$$

- For a metric d on partitions, we choose the "best" (optimal transport) coupling for this step of the joint chain

$$\phi^{\mathrm{OT}} = \underset{\text{couplings } \gamma}{\arg\min} \sum_k \sum_{k'} u_{k,k'} \mathrm{d}(\pi_k, \nu_{k'})$$

Coupling for partitions: our proposal

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- We propose to work directly in the partition space
- Given current partition, next is distributed:
  - 1st chain: $\sum_{k=1}^{K} a_k \delta_{\pi_k}$
  - 2nd chain: $\sum_{k'=1}^{K'} b_{k'} \delta_{\nu_{k'}}$
- Let coupling $\gamma = \sum_k \sum_{k'} u_{k,k'} \delta_{(\pi_k, \nu_{k'})}$ have marginals:

$$\sum_k u_{k,k'} = b_{k'} \quad \& \quad \sum_{k'} u_{k,k'} = a_k$$

- For a metric d on partitions, we choose the "best" (optimal transport) coupling for this step of the joint chain

$$\phi^{\mathrm{OT}} = \operatorname*{arg\,min}_{\text{couplings } \gamma} \sum_k \sum_{k'} u_{k,k'} \mathrm{d}(\pi_k, \nu_{k'}) \quad \text{with}$$

$$\mathrm{d}(\pi, \nu) = \sum_{A \in \pi} |A|^2 + \sum_{B \in \nu} |B|^2 - 2 \sum_{A \in \pi, B \in \nu} |A \cap B|^2$$

[Mirkin, Chernyi 1970; Rand 1971]

Coupling for partitions: our proposal

- Want: quickly compute unbiased $\hat{H} \approx H^* = \int h(X) p_X(X) dX$
  - Our case: $X = \Pi$ & e.g. $h(\Pi)$ is largest-cluster proportion
- We propose to work directly in the partition space
- Given current partition, next is distributed:
  - 1st chain: $\sum_{k=1}^{K} a_k \delta_{\pi_k}$
  - 2nd chain: $\sum_{k'=1}^{K'} b_{k'} \delta_{\nu_{k'}}$
- Let coupling $\gamma = \sum_k \sum_{k'} u_{k,k'} \delta_{(\pi_k, \nu_{k'})}$ have marginals:

$$\sum_k u_{k,k'} = b_{k'} \;\; \& \;\; \sum_{k'} u_{k,k'} = a_k$$

- For a metric d on partitions, we choose the "best" (optimal transport) coupling for this step of the joint chain

$$\phi^{\mathrm{OT}} = \underset{\text{couplings } \gamma}{\arg\min} \sum_k \sum_{k'} u_{k,k'} \mathrm{d}(\pi_k, \nu_{k'}) \quad \text{with}$$

$$\mathrm{d}(\pi, \nu) = \sum_{A \in \pi} |A|^2 + \sum_{B \in \nu} |B|^2 - 2 \sum_{A \in \pi, B \in \nu} |A \cap B|^2$$

[Mirkin, Chernyi 1970; Rand 1971]

- We use: network simplex method in Python Optimal Transport

[Flamary et al 2021]

Theory for our coupling proposal

Theory for our coupling proposal

- We focus on a Bayesian model: Dirichlet process (DP) mixture prior on the partition and Gaussian components

Theory for our coupling proposal

- We focus on a Bayesian model: Dirichlet process (DP) mixture prior on the partition and Gaussian components
- We focus on Gibbs sampling (and variants like split-merge)

Theory for our coupling proposal

- We focus on a Bayesian model: Dirichlet process (DP) mixture prior on the partition and Gaussian components
- We focus on Gibbs sampling (and variants like split-merge)
- **Theorem sketch (unbiasedness)**. The estimator $\hat{H}$ that arises from our optimal transport coupling is unbiased for H^*

Theory for our coupling proposal

- We focus on a Bayesian model: Dirichlet process (DP) mixture prior on the partition and Gaussian components
- We focus on Gibbs sampling (and variants like split-merge)
- **Theorem sketch (unbiasedness)**. The estimator $\hat{H}$ that arises from our optimal transport coupling is unbiased for H^*
    - *Proof sketch*. Check conditions from Jacob et al 2020: joint sampler is valid, meeting time has subgeometric tails, chains are faithful once they meet

Theory for our coupling proposal

- We focus on a Bayesian model: Dirichlet process (DP) mixture prior on the partition and Gaussian components
- We focus on Gibbs sampling (and variants like split-merge)
- **Theorem sketch (unbiasedness)**. The estimator $\hat{H}$ that arises from our optimal transport coupling is unbiased for H^*
  - *Proof sketch*. Check conditions from Jacob et al 2020: joint sampler is valid, meeting time has subgeometric tails, chains are faithful once they meet
- Take N = data cardinality, D = data dimension, K = largest number of partition components encountered

Theory for our coupling proposal

- We focus on a Bayesian model: Dirichlet process (DP) mixture prior on the partition and Gaussian components
- We focus on Gibbs sampling (and variants like split-merge)
- **Theorem sketch (unbiasedness)**. The estimator $\hat{H}$ that arises from our optimal transport coupling is unbiased for H^*
  - *Proof sketch*. Check conditions from Jacob et al 2020: joint sampler is valid, meeting time has subgeometric tails, chains are faithful once they meet
- Take *N* = data cardinality, *D* = data dimension, *K* = largest number of partition components encountered
- **Theorem sketch (time cost)**. A standard Gibbs step is $O(ND+KD^3)$. Computing our coupling costs 2 times a Gibbs step + optimal transport cost $O(K^3 \log K)$ (or K^2 or K^5)
  [Orlin 1993] [Bonneel et al 2011] [Kelly, O'Neill 1991, Sec 3.6]
  - In DP, we expect # clusters = $O(\log N)$

Theory for our coupling proposal

- We focus on a Bayesian model: Dirichlet process (DP) mixture prior on the partition and Gaussian components
- We focus on Gibbs sampling (and variants like split-merge)
- **Theorem sketch (unbiasedness)**. The estimator $\hat{H}$ that arises from our optimal transport coupling is unbiased for H^*
  - *Proof sketch*. Check conditions from Jacob et al 2020: joint sampler is valid, meeting time has subgeometric tails, chains are faithful once they meet
- Take N = data cardinality, D = data dimension, K = largest number of partition components encountered
- **Theorem sketch (time cost)**. A standard Gibbs step is O(ND+KD^3). Computing our coupling costs 2 times a Gibbs step + optimal transport cost O($K^3 \log K$) (or K^2 or K^5)
  [Orlin 1993] [Bonneel et al 2011] [Kelly, O'Neill 1991, Sec 3.6]
  - In DP, we expect # clusters = O(log$N$)
  - cf. meeting time

Experiments: estimator accuracy

Experiments: estimator accuracy

- When aggregating across processors, can either use a traditional mean or a trimmed mean.

Experiments: estimator accuracy

- When aggregating across processors, can either use a traditional mean or a trimmed mean. We compute % error.

Experiments: estimator accuracy

- When aggregating across processors, can either use a traditional mean or a trimmed mean. We compute % error.
  - Ground truth: MCMC run for really long

Experiments: estimator accuracy

- When aggregating across processors, can either use a traditional mean or a trimmed mean. We compute % error.
  - Ground truth: MCMC run for really long
  - Methods get same budget at each core (cf. cloud $)

Experiments: estimator accuracy

- When aggregating across processors, can either use a traditional mean or a trimmed mean. We compute % error.
  - Ground truth: MCMC run for really long
  - Methods get same budget at each core (cf. cloud $)
- We repeat whole procedure many times (unlike in practice) to assess variability & report 20, 50, 80% quantiles

Experiments: estimator accuracy

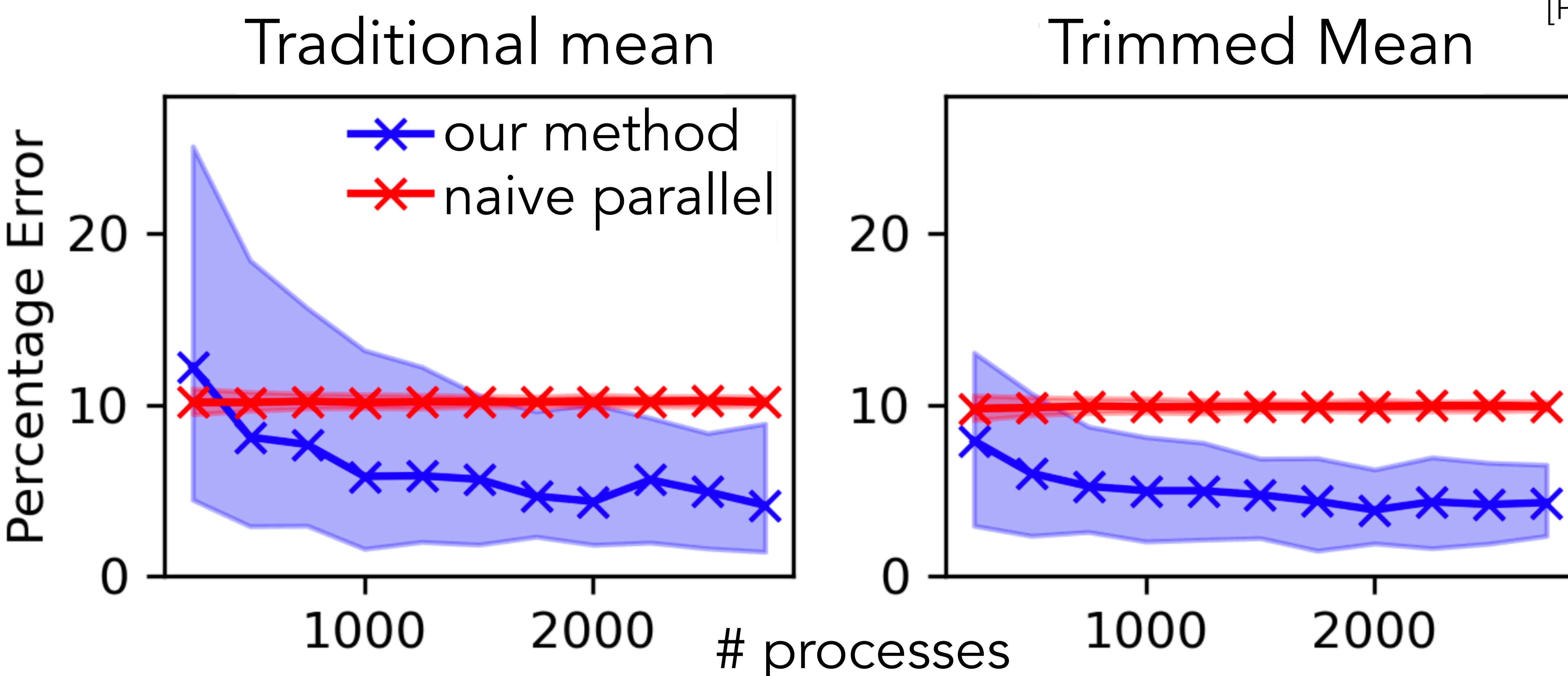
- When aggregating across processors, can either use a traditional mean or a trimmed mean. We compute % error.
  - Ground truth: MCMC run for really long
  - Methods get same budget at each core (cf. cloud $)
- We repeat whole procedure many times (unlike in practice) to assess variability & report 20, 50, 80% quantiles
- Single-cell RNA seq data, h=largest component proportion

[Prabhakaran et al 2016]

Experiments: estimator accuracy

- When aggregating across processors, can either use a traditional mean or a trimmed mean. We compute % error.
  - Ground truth: MCMC run for really long
  - Methods get same budget at each core (cf. cloud $)
- We repeat whole procedure many times (unlike in practice) to assess variability & report 20, 50, 80% quantiles
- Single-cell RNA seq data, *h*=largest component proportion

[Prabhakaran et al 2016]



Experiments: estimator accuracy

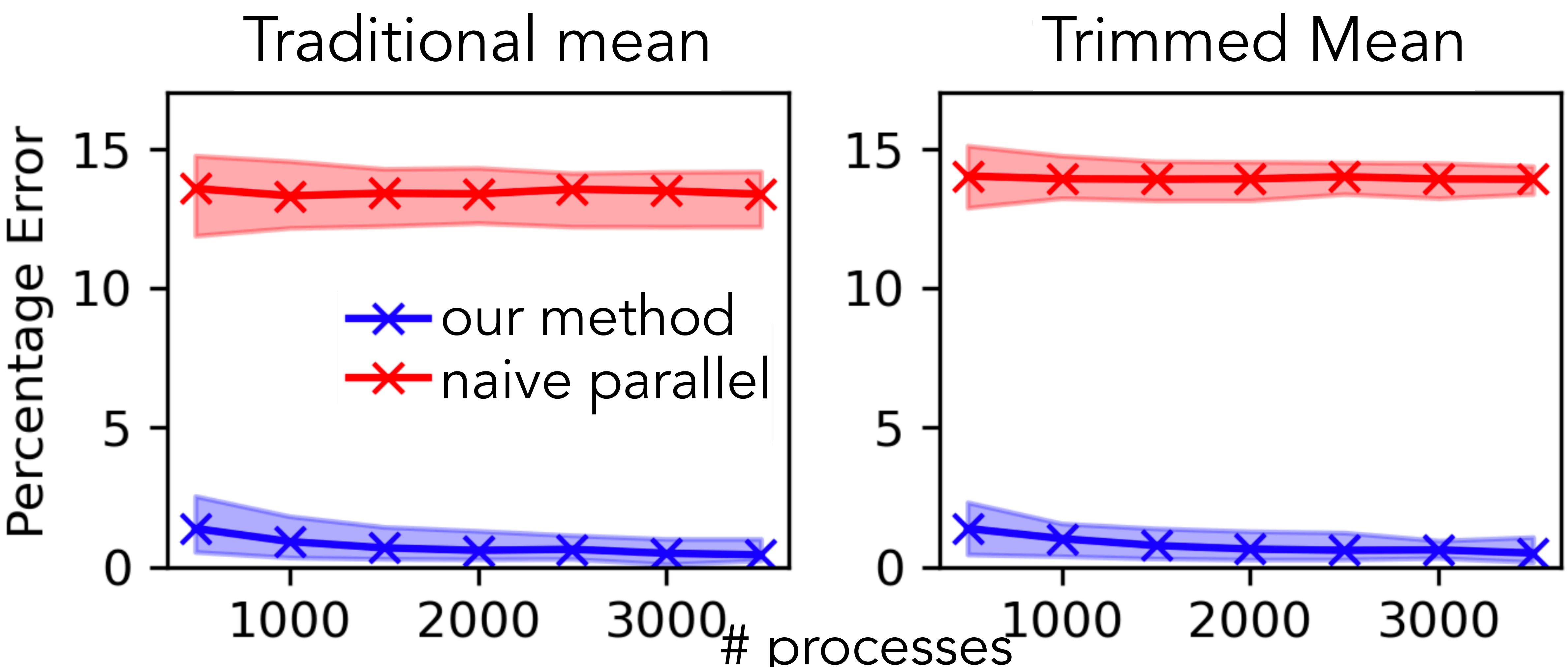
- When aggregating across processors, can either use a traditional mean or a trimmed mean. We compute % error.
  - Ground truth: MCMC run for really long
  - Methods get same budget at each core (cf. cloud $)
- We repeat whole procedure many times (unlike in practice) to assess variability & report 20, 50, 80% quantiles

Experiments: estimator accuracy

- When aggregating across processors, can either use a traditional mean or a trimmed mean. We compute % error.
  - Ground truth: MCMC run for really long
  - Methods get same budget at each core (cf. cloud $)
- We repeat whole procedure many times (unlike in practice) to assess variability & report 20, 50, 80% quantiles
- Graph colorings, h=co-clustering probability

Experiments: estimator accuracy

- When aggregating across processors, can either use a traditional mean or a trimmed mean. We compute % error.
  - Ground truth: MCMC run for really long
  - Methods get same budget at each core (cf. cloud $)
- We repeat whole procedure many times (unlike in practice) to assess variability & report 20, 50, 80% quantiles
- Graph colorings, *h*=co-clustering probability

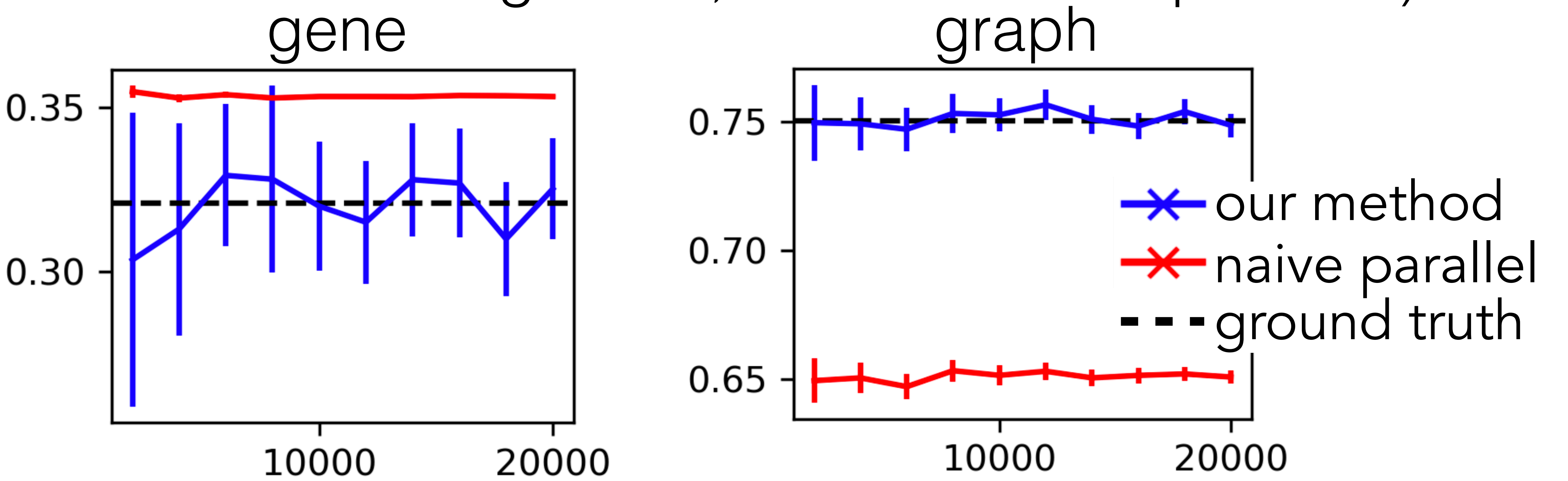


Experiments

- Want to assess the quality of confidence intervals (formed from one run of the algorithm, as we would in practice)

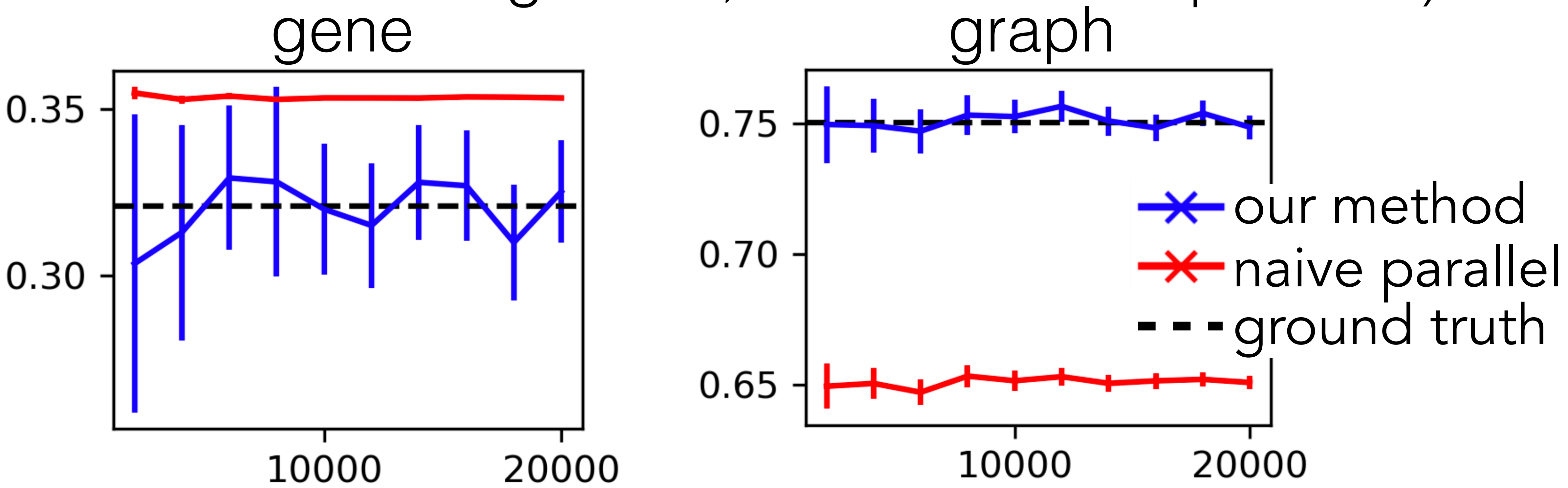
Experiments

- Want to assess the quality of confidence intervals (formed from one run of the algorithm, as we would in practice)



Experiments

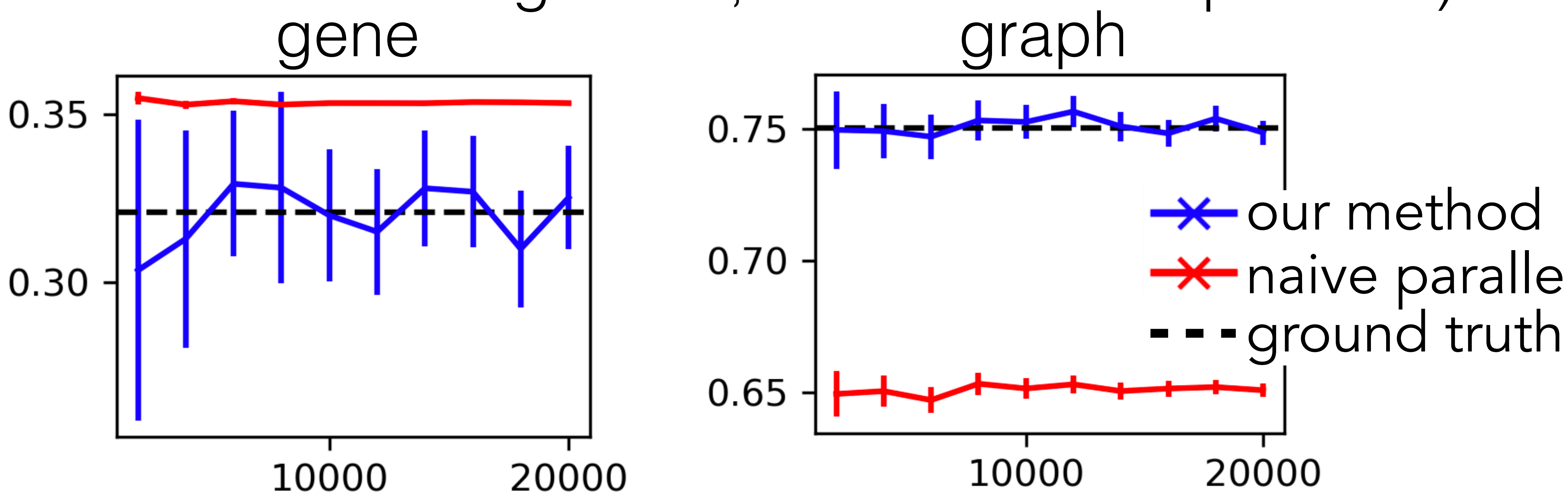
- Want to assess the quality of confidence intervals (formed from one run of the algorithm, as we would in practice)



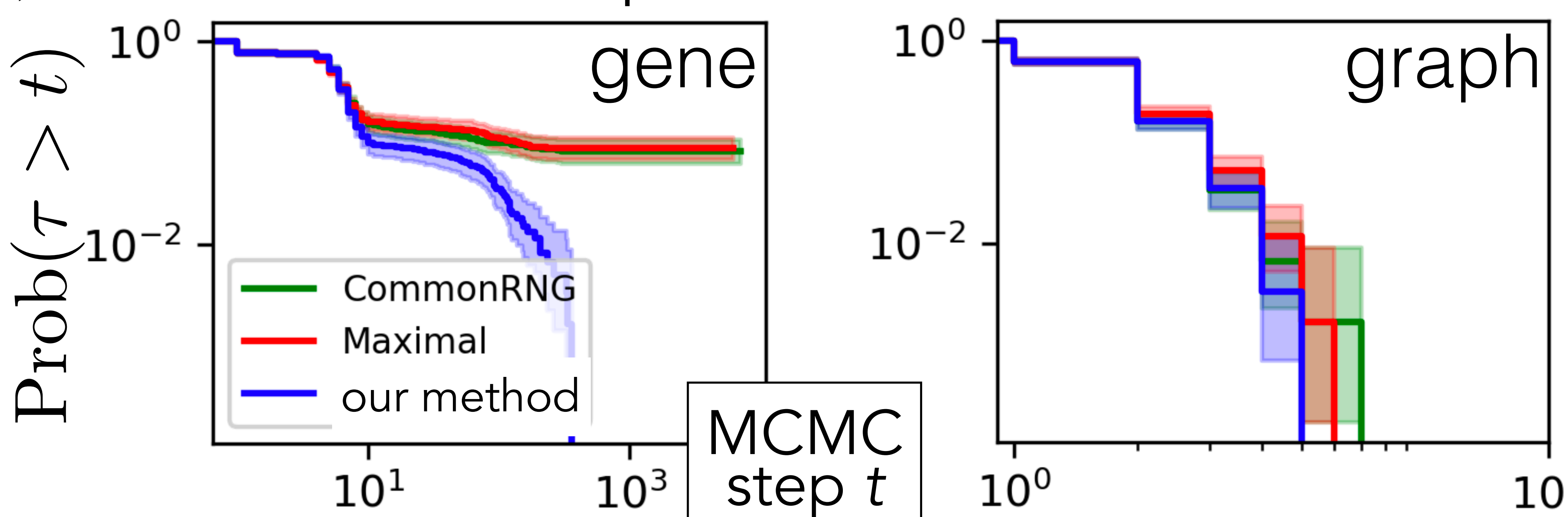
- Label-based couplings don't meet in the time budget of our experiments, so we didn't compare with them above

Experiments

- Want to assess the quality of confidence intervals (formed from one run of the algorithm, as we would in practice)

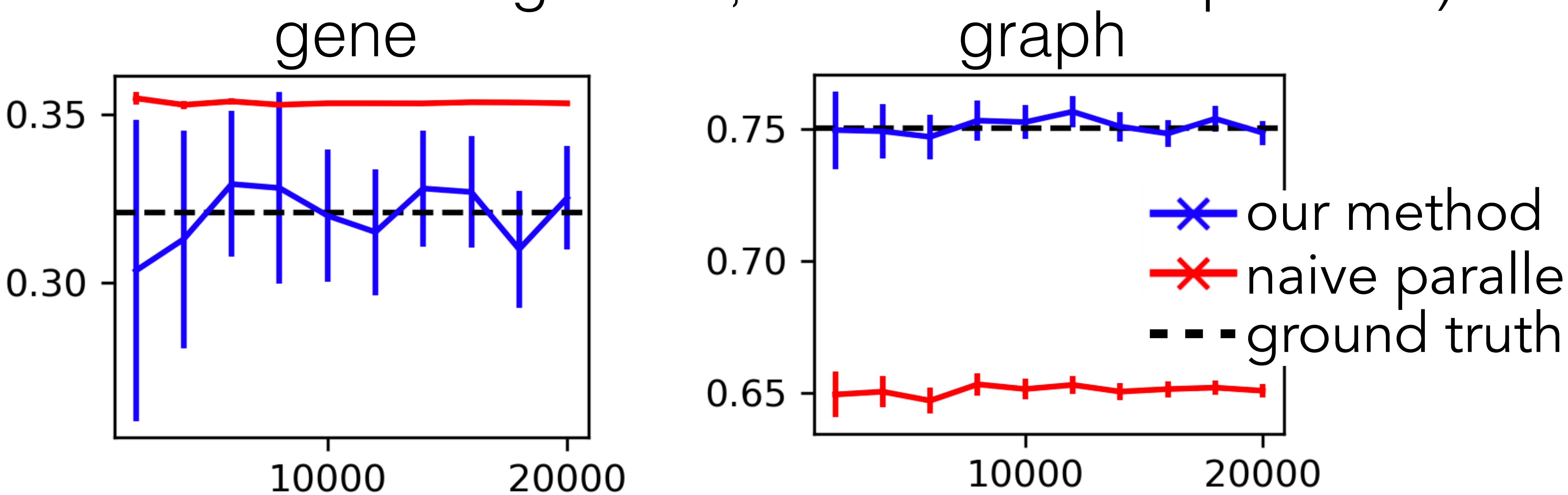


- Label-based couplings don’t meet in the time budget of our experiments, so we didn’t compare with them above



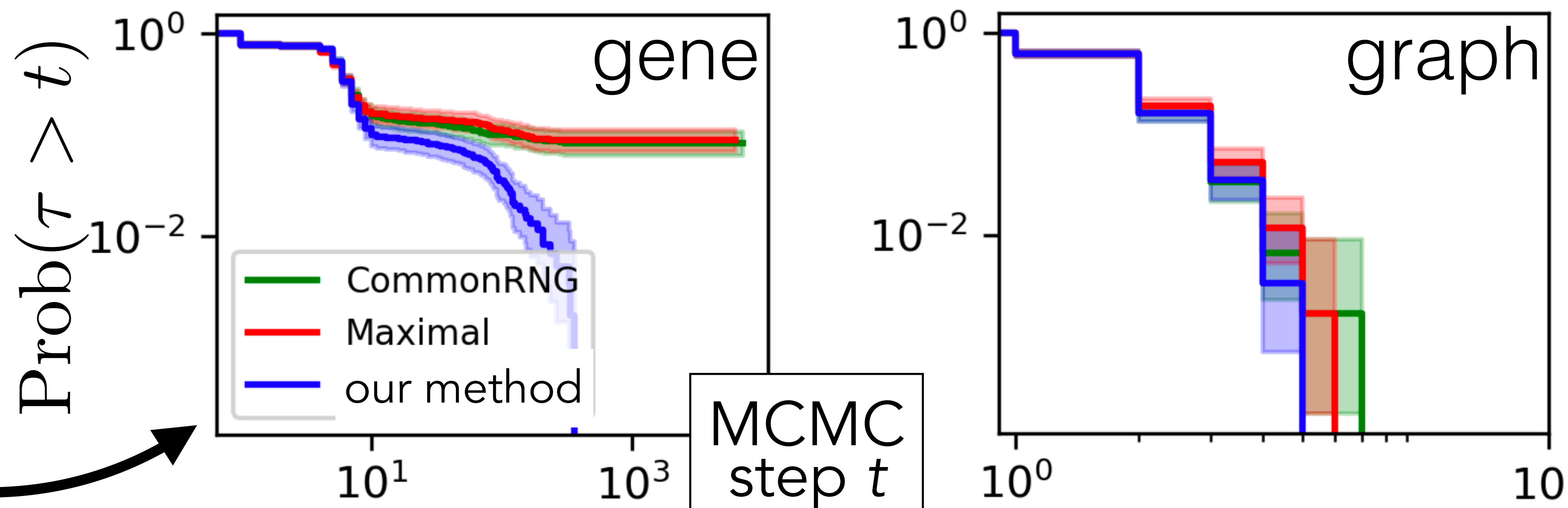
Experiments

- Want to assess the quality of confidence intervals (formed from one run of the algorithm, as we would in practice)



- Label-based couplings don't meet in the time budget of our experiments, so we didn't compare with them above

same behavior in abalone, seed, and synthetic



Conclusions

- We provide an optimal transport coupling for partitions
  - In the highly parallel regime, we show our coupling can provide accurate estimates with limited wall time
- **Paper**: Nguyen, Trippe, Broderick “Many Processors, Little Time: MCMC for Partitions via Optimal Transport Couplings” *AISTATS* 2022.
- **Code**: `https://github.com/tinnguyen96/partition-coupling`

Conclusions

- We provide an optimal transport coupling for partitions
  - In the highly parallel regime, we show our coupling can provide accurate estimates with limited wall time
- **Paper**: Nguyen, Trippe, Broderick "Many Processors, Little Time: MCMC for Partitions via Optimal Transport Couplings" *AISTATS* 2022.
- **Code**: `https://github.com/tinnguyen96/partition-coupling`
- Additional thoughts:

Conclusions

- We provide an optimal transport coupling for partitions
  - In the highly parallel regime, we show our coupling can provide accurate estimates with limited wall time
- **Paper**: Nguyen, Trippe, Broderick "Many Processors, Little Time: MCMC for Partitions via Optimal Transport Couplings" *AISTATS* 2022.
- **Code**: `https://github.com/tinnguyen96/partition-coupling`
- Additional thoughts:
  - Expect ideas to extend to pretty generic clustering but also other unsupervised learning: topic/feature models, trait models, etc.

Conclusions

- We provide an optimal transport coupling for partitions
  - In the highly parallel regime, we show our coupling can provide accurate estimates with limited wall time
- **Paper**: Nguyen, Trippe, Broderick “Many Processors, Little Time: MCMC for Partitions via Optimal Transport Couplings” *AISTATS* 2022.
- **Code**: `https://github.com/tinnguyen96/partition-coupling`
- Additional thoughts:
  - Expect ideas to extend to pretty generic clustering but also other unsupervised learning: topic/feature models, trait models, etc.
  - Optimal transport for couplings in continuous problems: Xu et al “Couplings for multinomial Hamiltonian Monte Carlo” *AISTATS* 2021