Distributed Learning of Finite Gaussian Mixtures

Qiong Zhang
Institute of Statistics and Big Data, RUC

IASM-BIRS workshop: Harnessing the power of latent structure models and modern big data learning

Hangzhou, China
December 14, 2023

Joint work with Dr. Jiahua Chen

Zhang, Q., & Chen, J. (2022). Distributed learning of finite gaussian mixtures. *The Journal of Machine Learning Research*, *23*(1), 4265-4304.

Finite mixture models

- A family of distributions.
- Let $\mathscr{F} = \{f(x;\theta) : \theta \in \Theta\}$ be a parametric family.
- The finite mixture model of $\mathscr{F}$ has it density function:

$$f(x;G) := \int f(x;\theta)\, dG(\theta) = \sum_{k=1}^{K} w_k f(x;\theta_k)$$

Finite mixture models

- A family of distributions.
- Let $\mathscr{F} = \{f(x;\theta) : \theta \in \Theta\}$ be a parametric family.
- The finite mixture model of $\mathscr{F}$ has it density function:

$$f(x;\boxed{G}) := \int f(x;\theta)\,dG(\theta) = \sum_{k=1}^{K} w_k f(x;\theta_k)$$

Mixing distribution

$$G = \sum_k w_k \delta_{\theta_k}$$

Finite mixture models

- A family of distributions.
- Let $\mathscr{F} = \{f(x;\theta) : \theta \in \Theta\}$ be a parametric family.
- The finite mixture model of $\mathscr{F}$ has it density function:

$$f(x;\boxed{G}) := \int f(x;\theta)\,dG(\theta) = \sum_{k=1}^{K} w_k f(x;\boxed{\theta_k})$$

Mixing distribution

Subpopulation parameter

$$G = \sum_k \boxed{w_k}\delta_{\theta_k}$$

Mixing weight

Finite mixture models

- A family of distributions.
- Let $\mathscr{F} = \{f(x;\theta) : \theta \in \Theta\}$ be a parametric family.
- The finite mixture model of $\mathscr{F}$ has it density function:

Order (assumed to be known)

$$f(x;\boxed{G}) := \int f(x;\theta)\,dG(\theta) = \sum_{k=1}^{\boxed{K}} w_k f(x;\boxed{\theta_k})$$

Mixing distribution

Subpopulation parameter

$$G = \sum_k \boxed{w_k}\delta_{\theta_k}$$

Mixing weight

Finite mixture models

- A family of distributions.
- Let $\mathscr{F} = \{f(x;\theta) : \theta \in \Theta\}$ be a parametric family.
- The finite mixture model of $\mathscr{F}$ has it density function:

Order (assumed to be known)

$$f(x;\boxed{G}) := \int f(x;\theta)\,dG(\theta) = \sum_{k=1}^{\boxed{K}} w_k f(x;\boxed{\theta_k})$$

Mixing distribution

Subpopulation parameter

$$G = \sum_k \boxed{w_k}\delta_{\theta_k}$$

Mixing weight

Finite Gaussian Mixture

$$\mathscr{F} = \{\phi(x;\mu,\Sigma) = |2\pi\Sigma|^{-1/2}\exp\{-(x-\mu)^\top\Sigma^{-1}(x-\mu)/2\} : \mu \in \mathbb{R}^d, \Sigma > 0\}$$

PDF $\phi(x;G)$ CDF $\Phi(x;G)$

Reason to parameterize by G

Consider the 2-component mixture

$$\phi(x; G) = 0.4\phi(x; -1,2) + 0.6\phi(x; 1,1)$$

- One may want to use a vector such as
$$\xi = (0.4, -1,2,0.6,1,1)$$
to parametrize the mixture
- Such parameterization may lead to unidentifiable model
  - Let $\xi_1 = (0.4, -1,2,0.6,1,1)$ and $\xi_2 = (0.6,1,1,0.4, -1,2)$
  - Note $\xi_1 \neq \xi_2$ but $\phi(x; \xi_1) = \phi(x; \xi_2)$
- The mixing distribution G does not have this issue

Finite mixture model in machine learning

Clustering

Latent variable representation

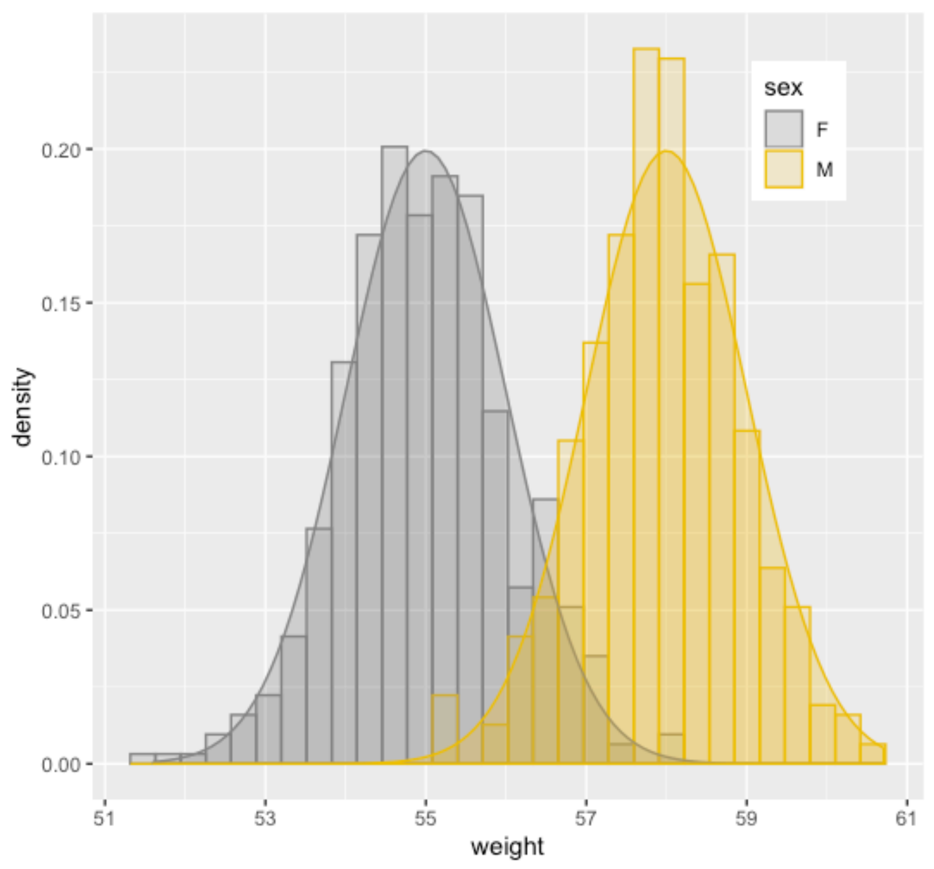
$$\begin{cases} X \mid Z = k \sim f(x; \theta_k) \\ P(Z = k) = w_k \end{cases}$$

Posterior distribution of the latent variable

$$P(Z = k \mid X = x) \propto w_k f(x; \theta_k)$$

Clustering

$$\kappa(x; G) = \text{argmax}_{j \in [K]} w_j f(x; \theta_j)$$



Finite mixture model in machine learning

Clustering

Latent variable representation

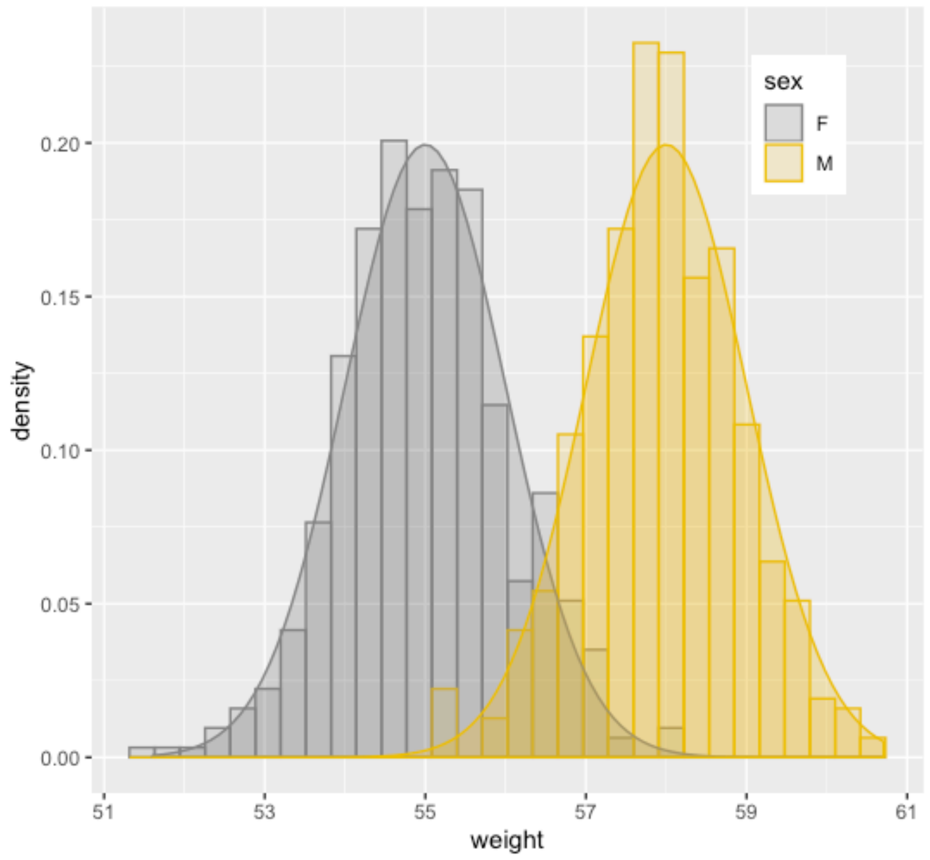
$$\begin{cases} X \mid Z = k \sim f(x; \theta_k) \\ P(Z = k) = w_k \end{cases}$$

Posterior distribution of the latent variable

$$P(Z = k \mid X = x) \propto w_k f(x; \theta_k)$$

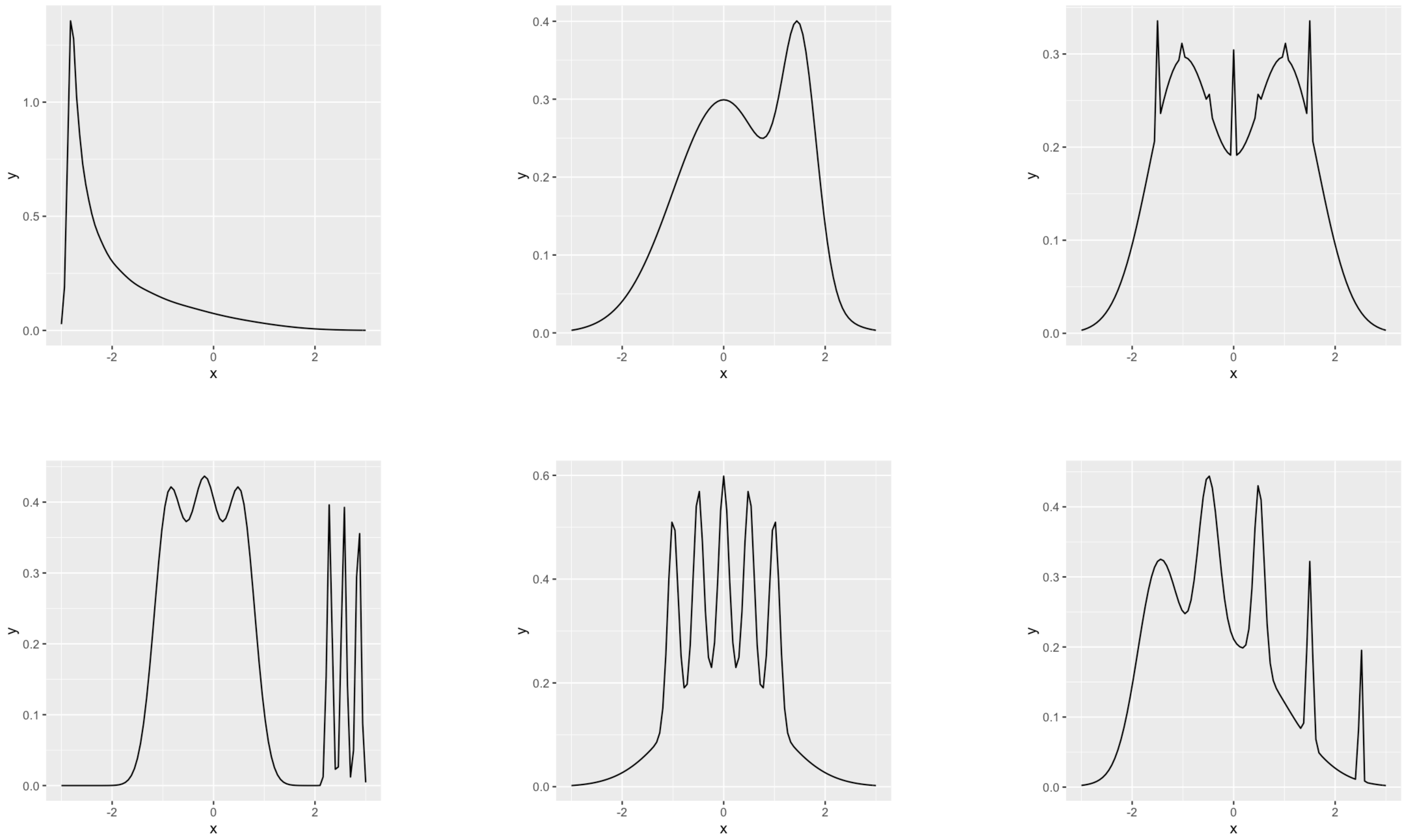
Clustering

$$\kappa(x; G) = \text{argmax}_{j \in [K]} w_j f(x; \theta_j)$$



Density Approximation

A parametric model that approximates density functions with various shapes



Density function credit: Geoffrey McLachlan and David Peel — Finite Mixture Models

Finite mixture model in machine learning

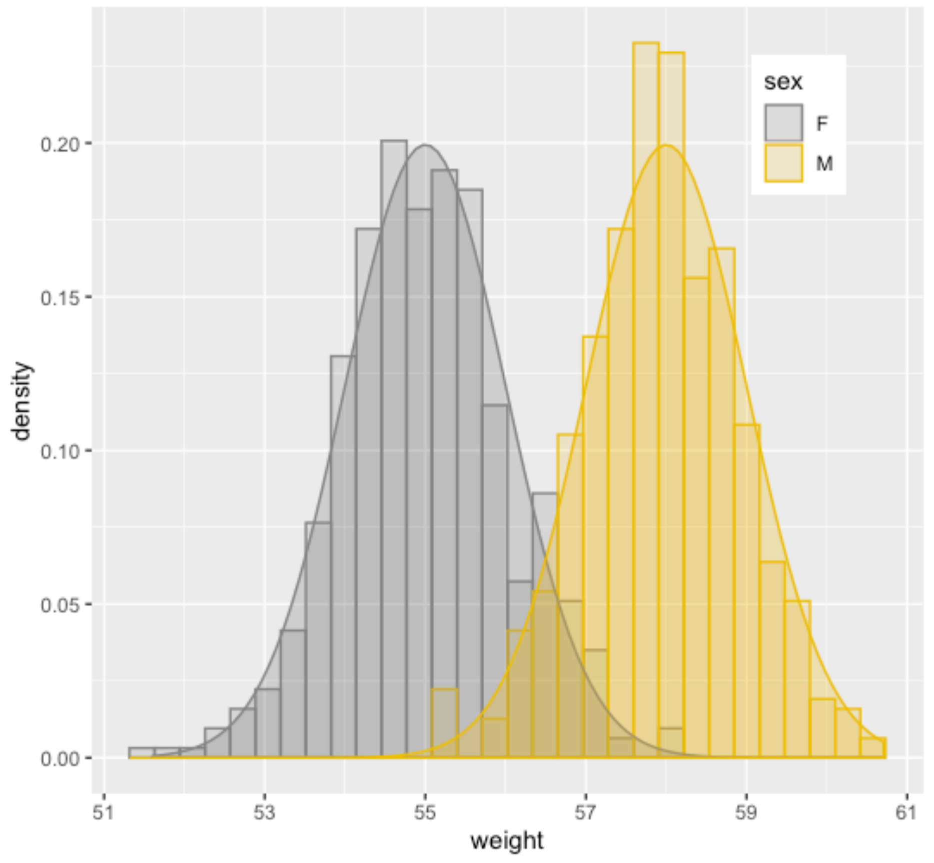
Clustering

Latent variable representation

$$\begin{cases} X \mid Z = k \sim f(x; \theta_k) \\ P(Z = k) = w_k \end{cases}$$

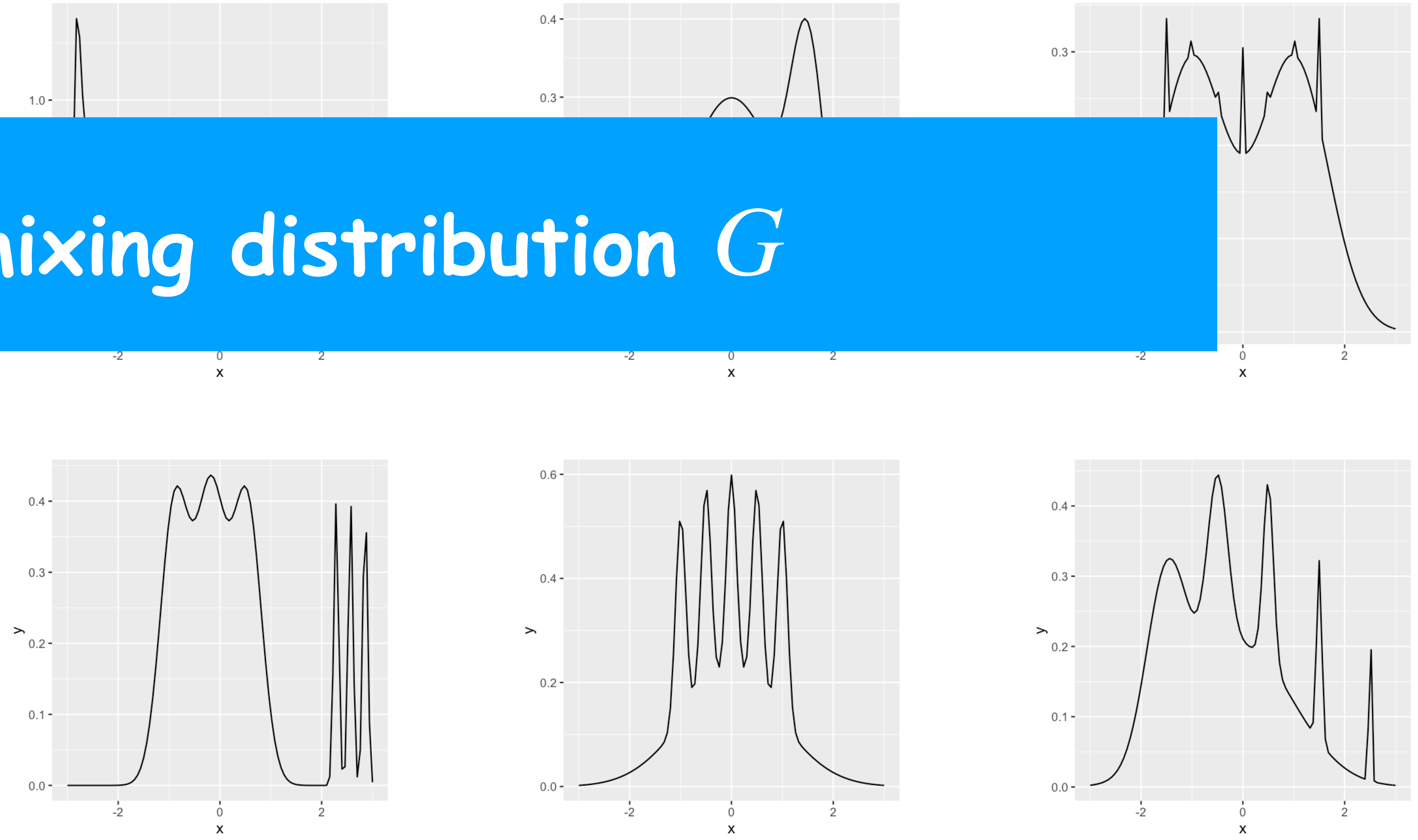
Posterior distribution of the latent variable

Clustering



Density Approximation

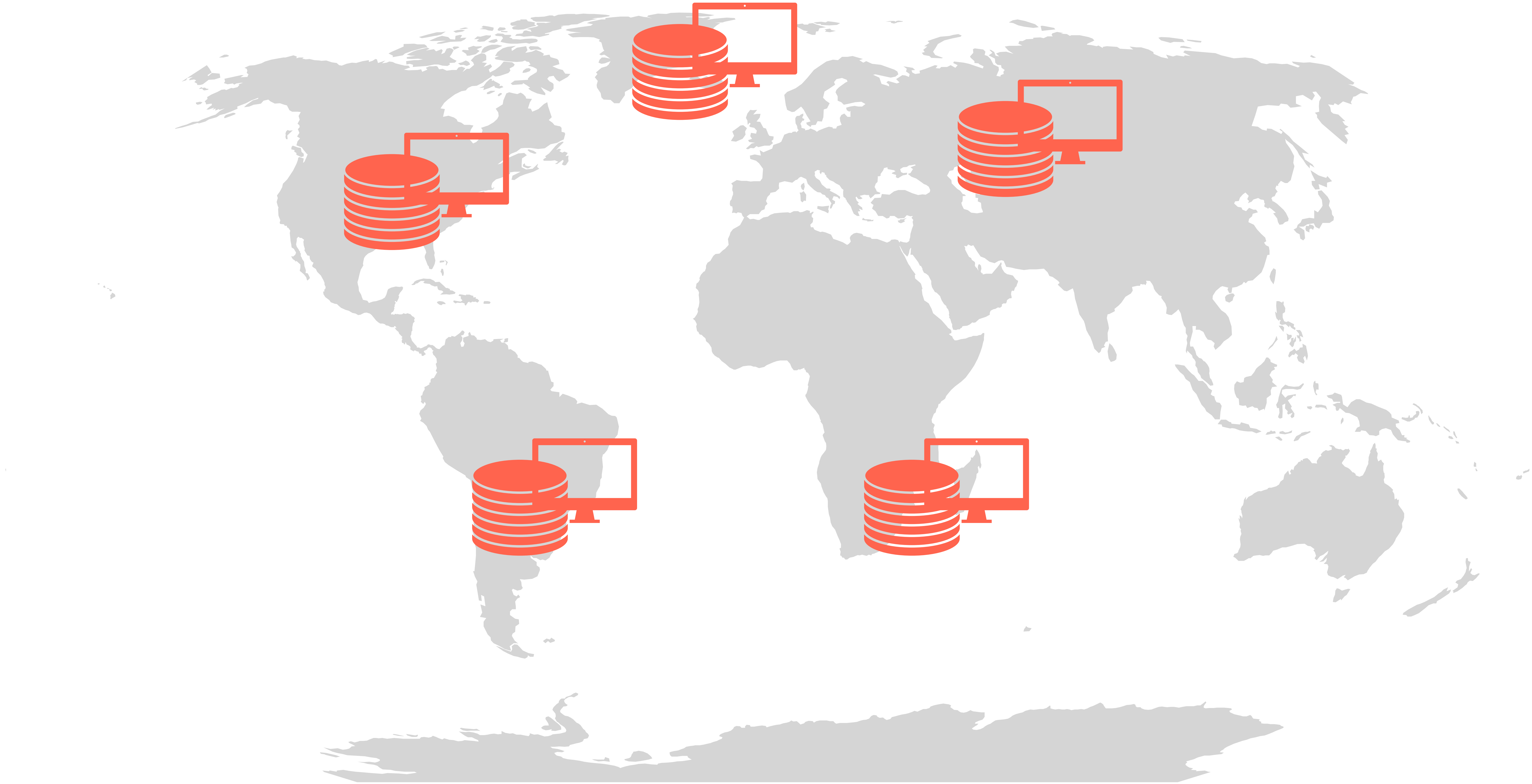
A parametric model that approximates density functions with various shapes



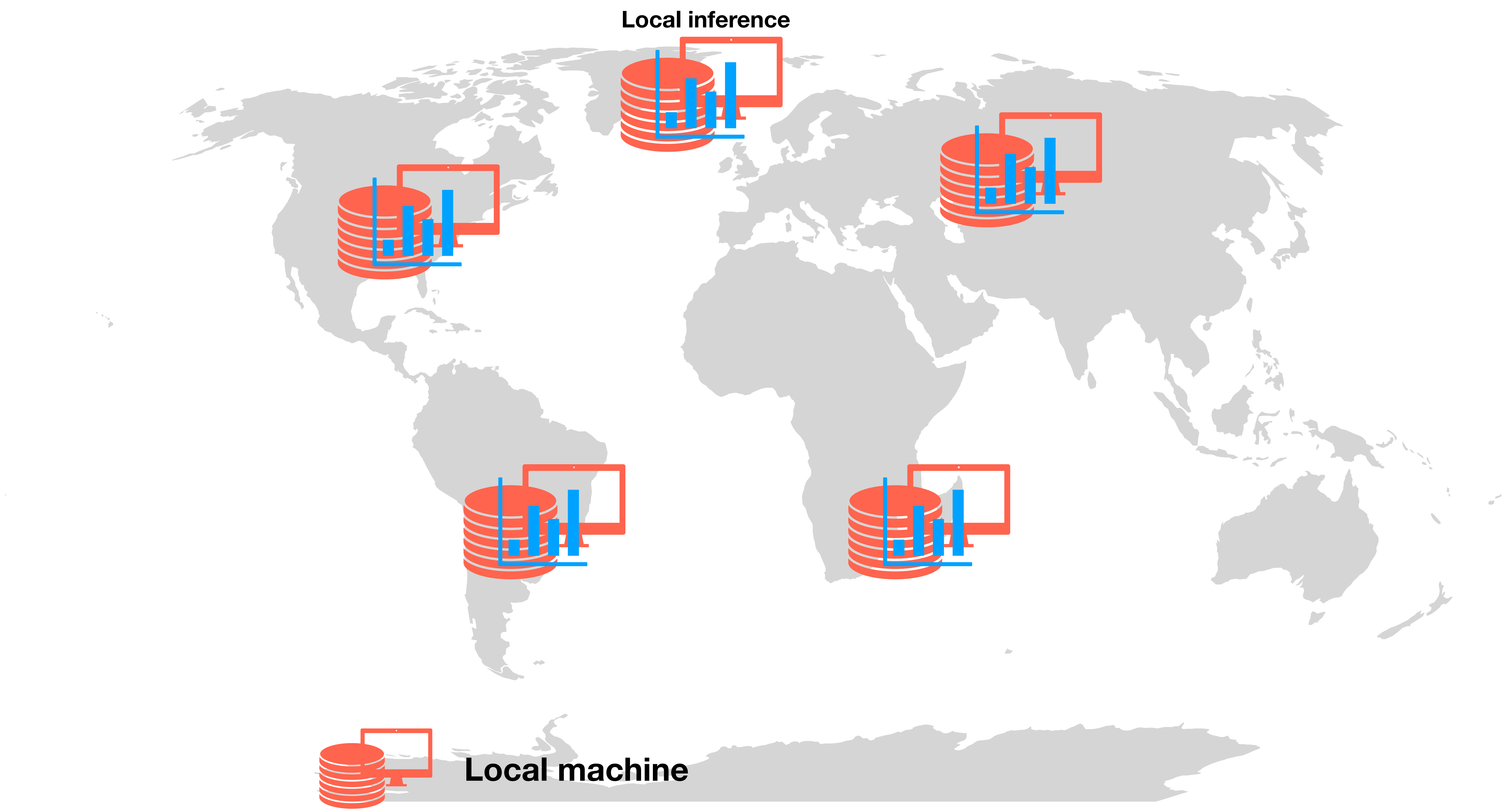
Goal: learn mixing distribution G

Density function credit: Geoffrey McLachlan and David Peel — Finite Mixture Models

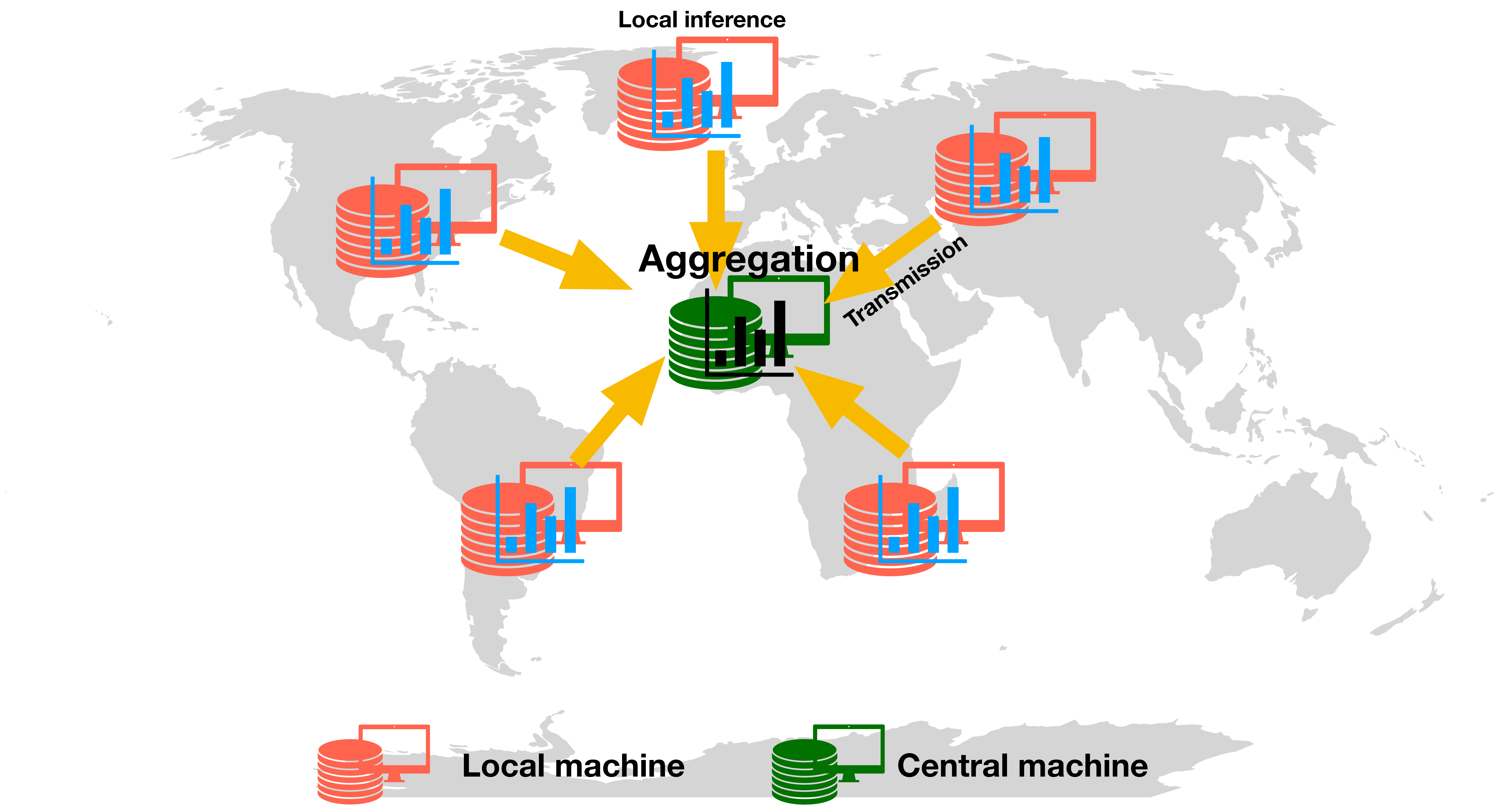
Split-and-conquer



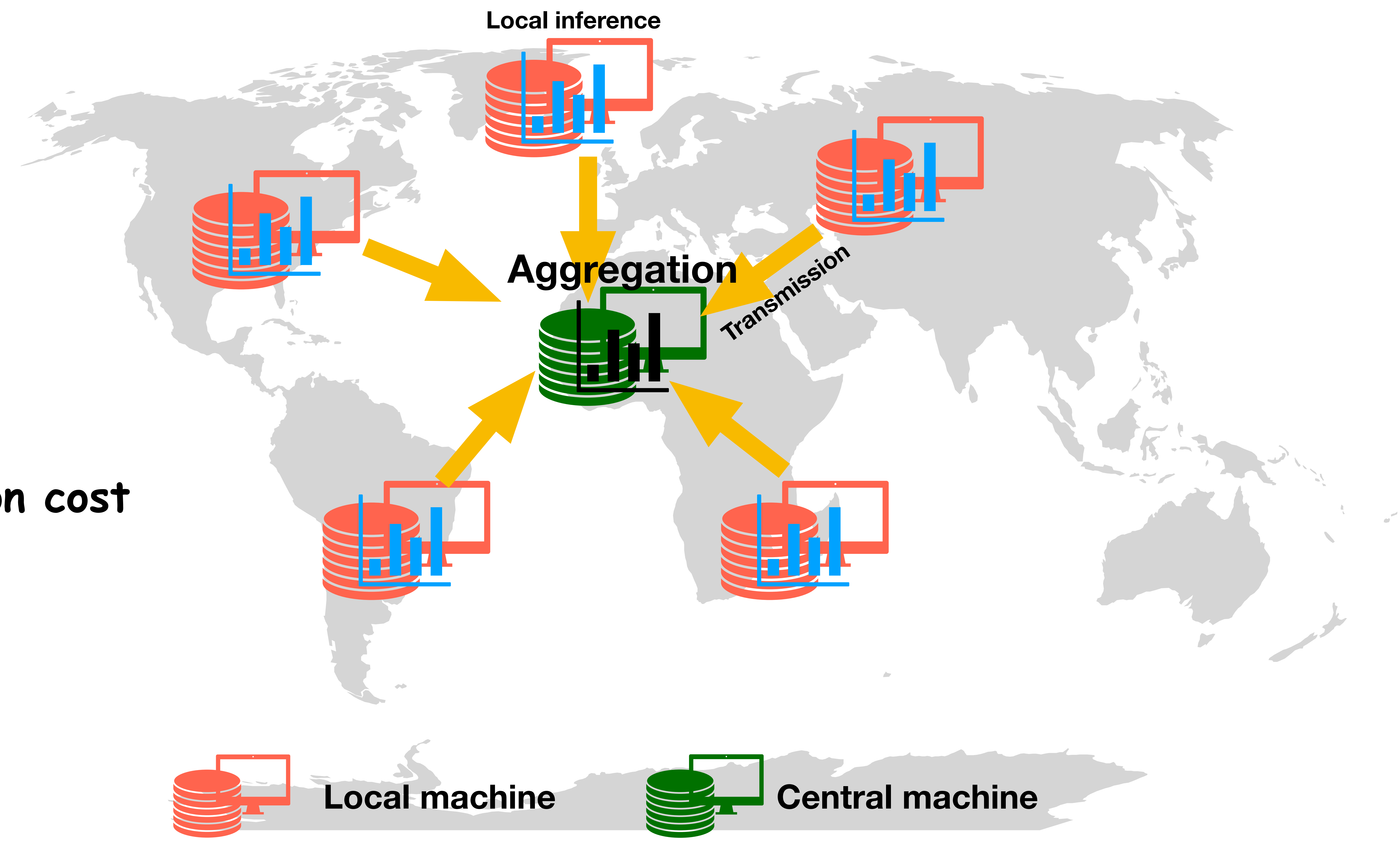
Split-and-conquer



Split-and-conquer



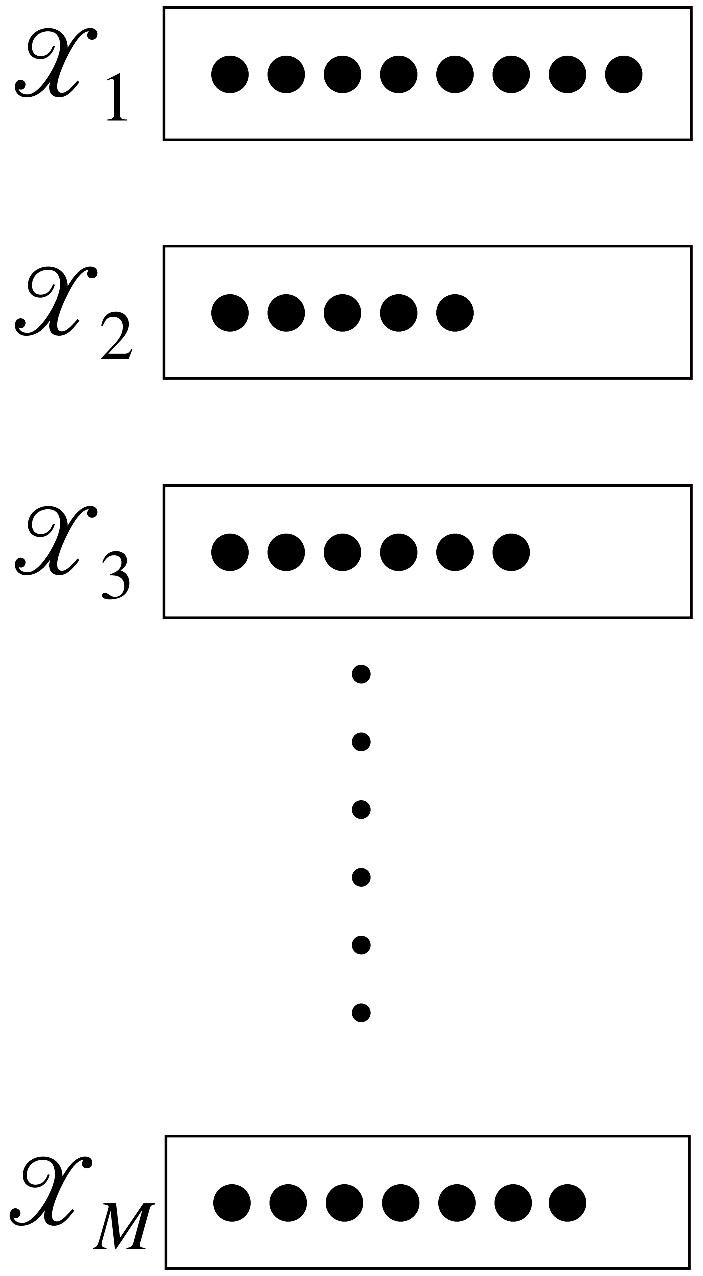
Split-and-conquer



- ✓ Privacy gain
- ✓ Low transmission cost

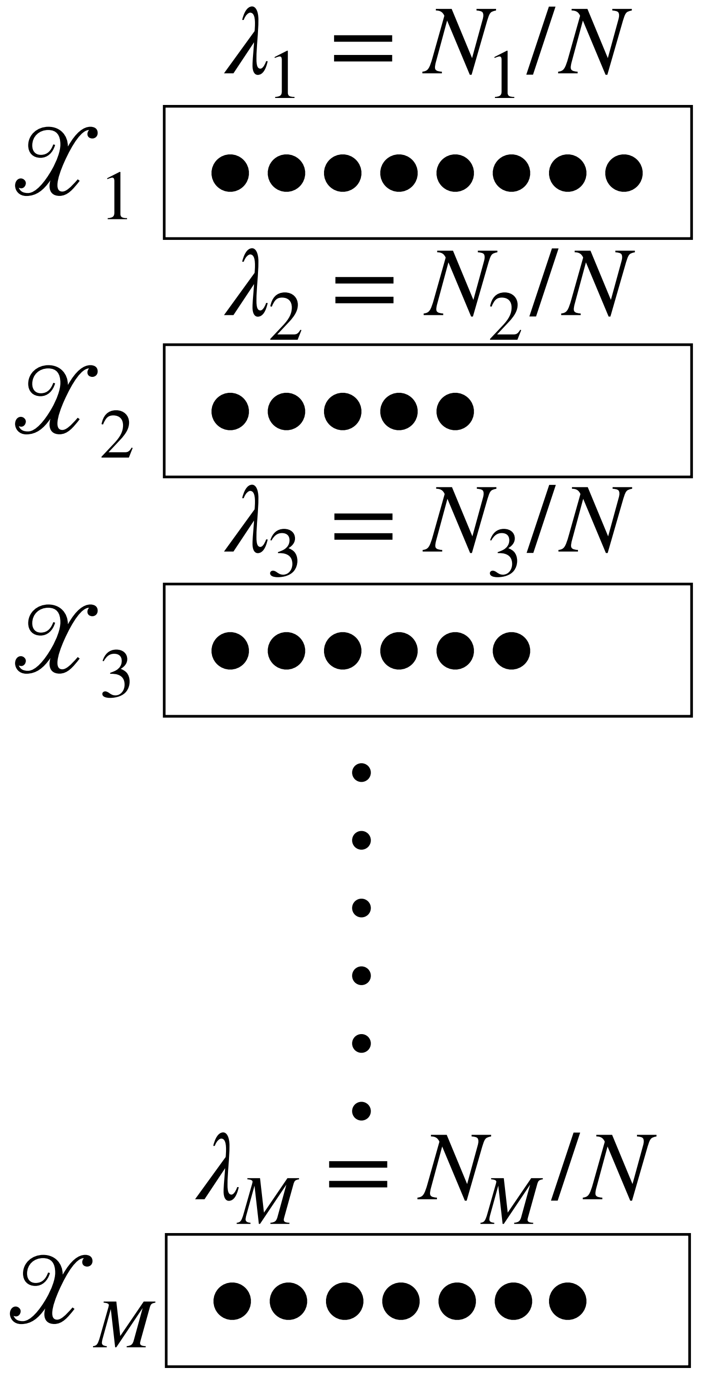
Split-and-conquer under Gaussian mixtures

Local datasets



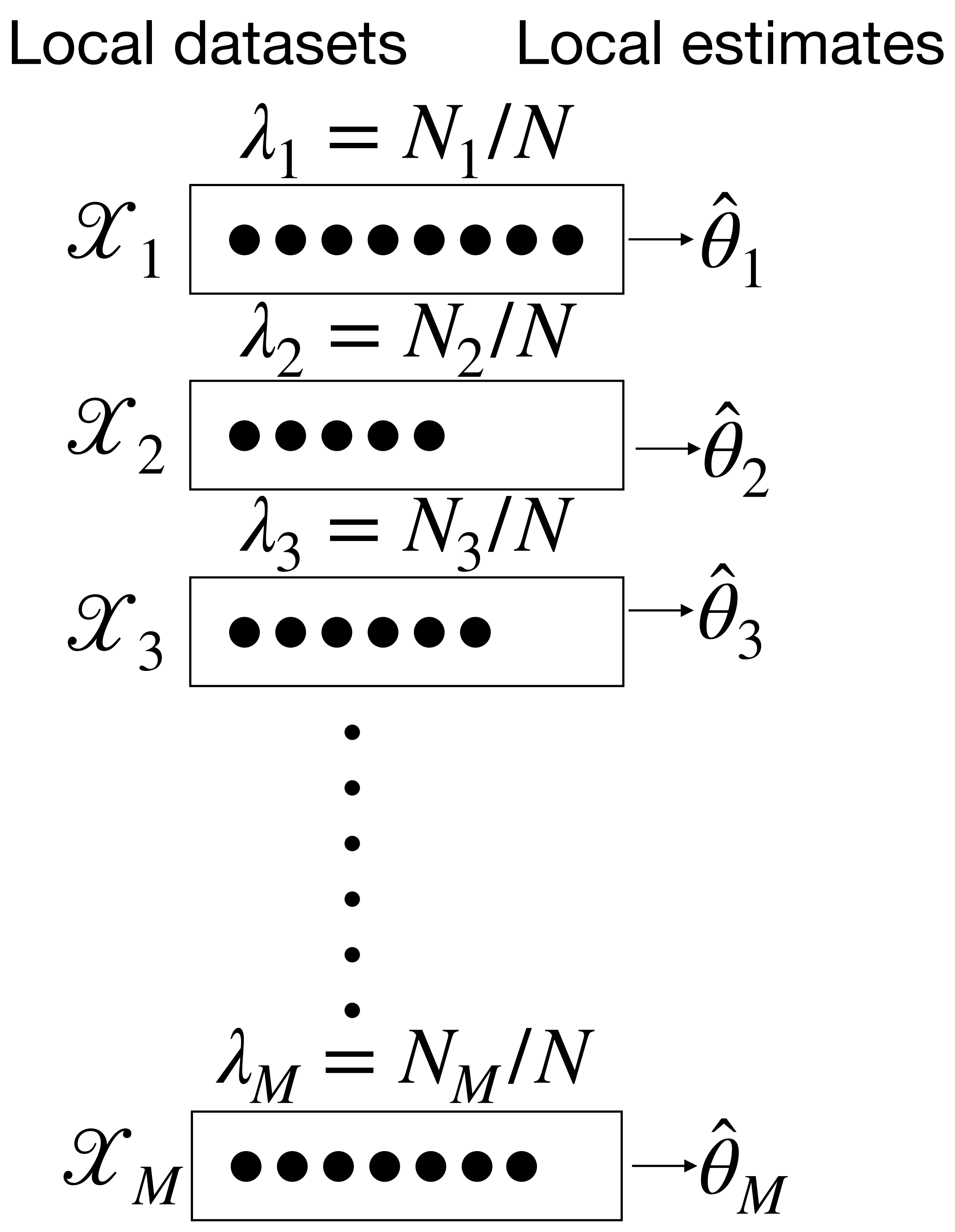
Split-and-conquer under Gaussian mixtures

Local datasets



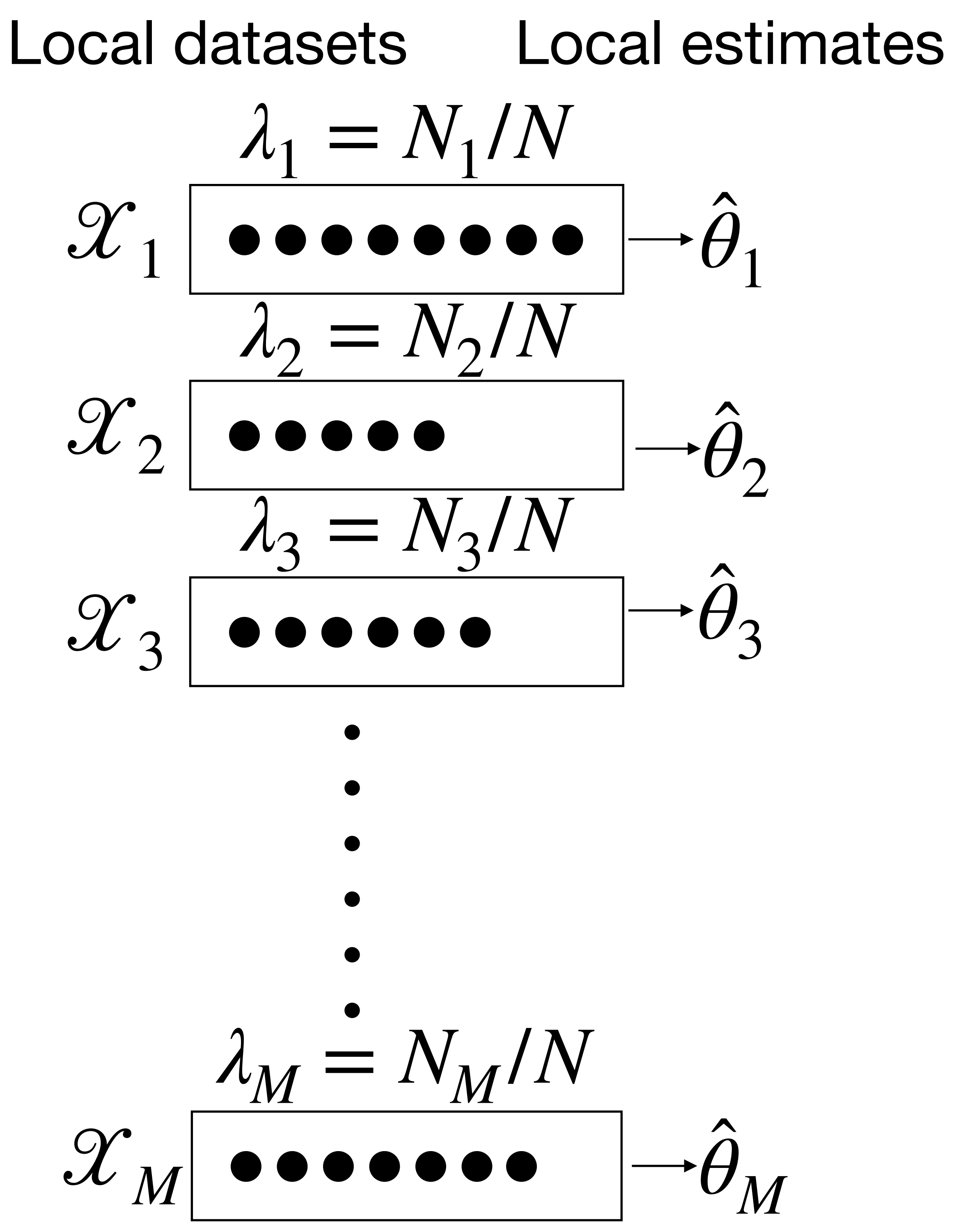
IID observations from $f(x; \theta^*)$

Split-and-conquer under Gaussian mixtures



IID observations from $f(x; \theta^*)$

Split-and-conquer under Gaussian mixtures

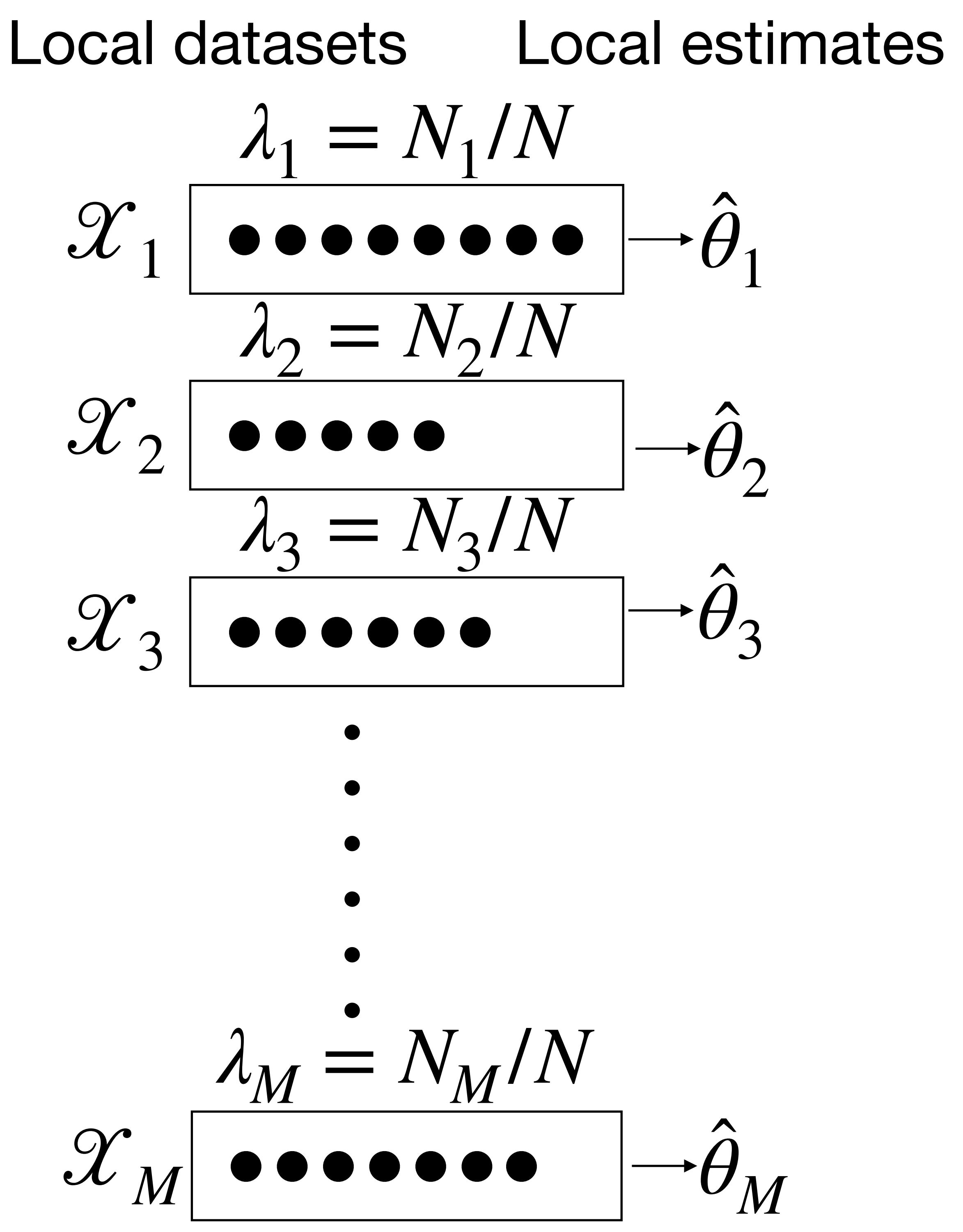


IID observations from $f(x; \theta^*)$

- Aggregation for real valued parameters:

$$\bar{\theta} = \sum_{m=1}^{M} \lambda_m \hat{\theta}_m$$

Split-and-conquer under Gaussian mixtures



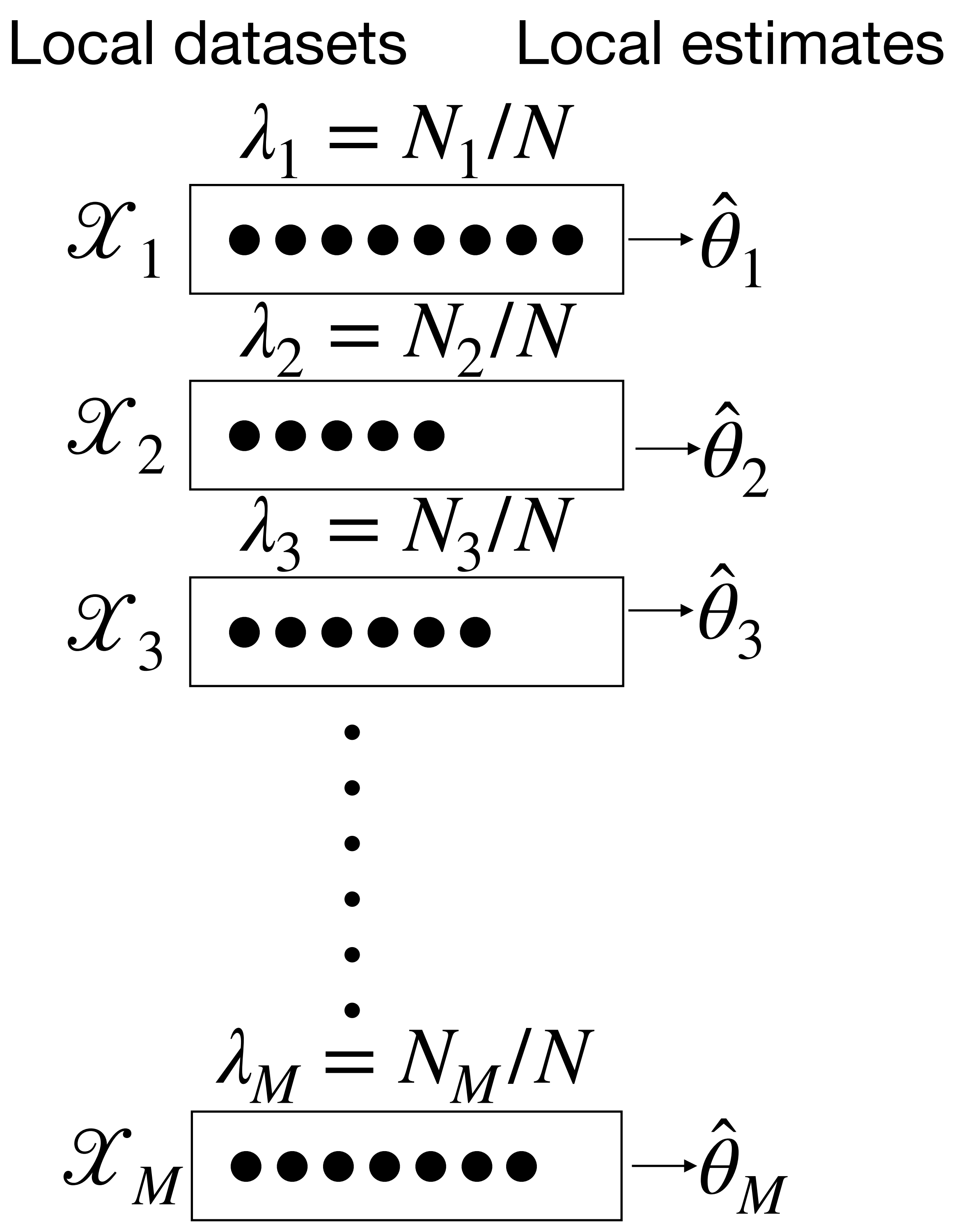
IID observations from $f(x; \theta^*)$

- Aggregation for real valued parameters:

$$\bar{\theta} = \sum_{m=1}^{M} \lambda_m \hat{\theta}_m$$

- Under GMM:
    - Parameter space is formed by discrete distributions with K support points.

Split-and-conquer under Gaussian mixtures



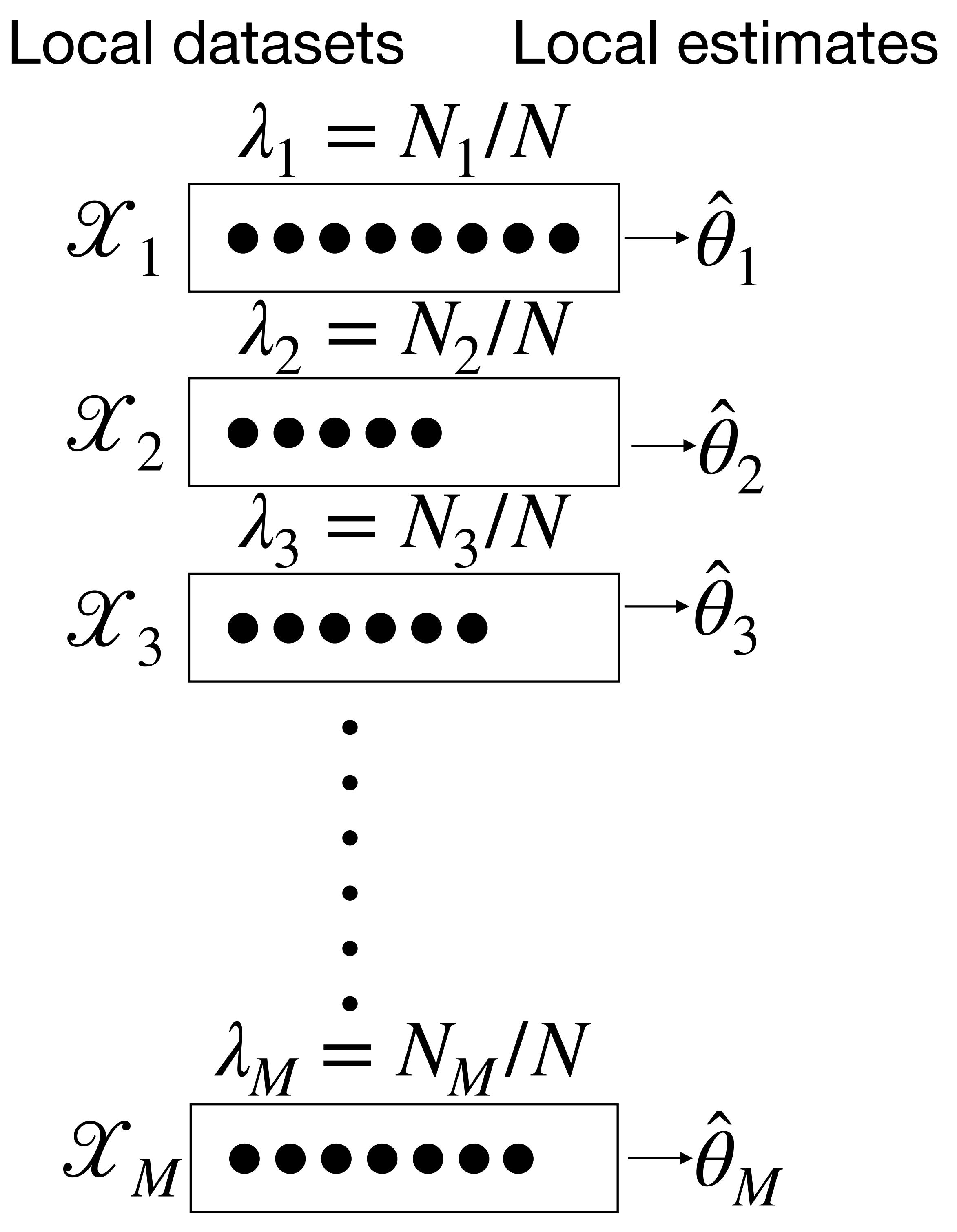
IID observations from $f(x; \theta^*)$

- Aggregation for real valued parameters:

$$\bar{\theta} = \sum_{m=1}^{M} \lambda_m \hat{\theta}_m$$

- Under GMM:
  - Parameter space is formed by discrete distributions with K support points.
  - Let $\bar{G} = \sum \lambda_m \hat{G}_m$

Split-and-conquer under Gaussian mixtures



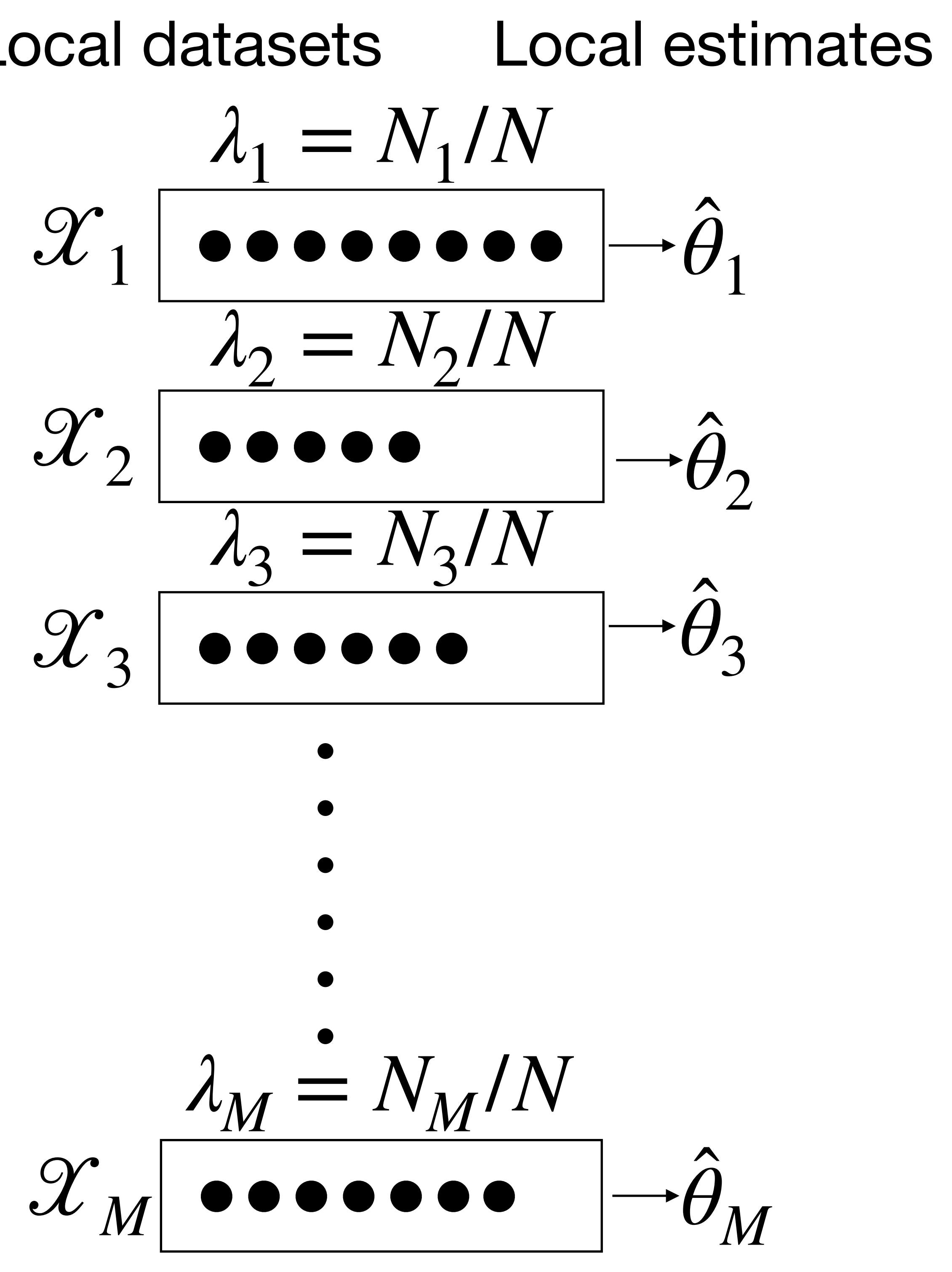
IID observations from $f(x; \theta^*)$

- Aggregation for real valued parameters:

$$\bar{\theta} = \sum_{m=1}^{M} \lambda_m \hat{\theta}_m$$

- Under GMM:
  - Parameter space is formed by discrete distributions with K support points.
  - Let $\bar{G} = \sum \lambda_m \hat{G}_m$
  - **Average mixture:** $\phi(x; \bar{G}) = \sum \lambda_m \phi(x; \hat{G}_m)$

Split-and-conquer under Gaussian mixtures



IID observations from $f(x; \theta^*)$

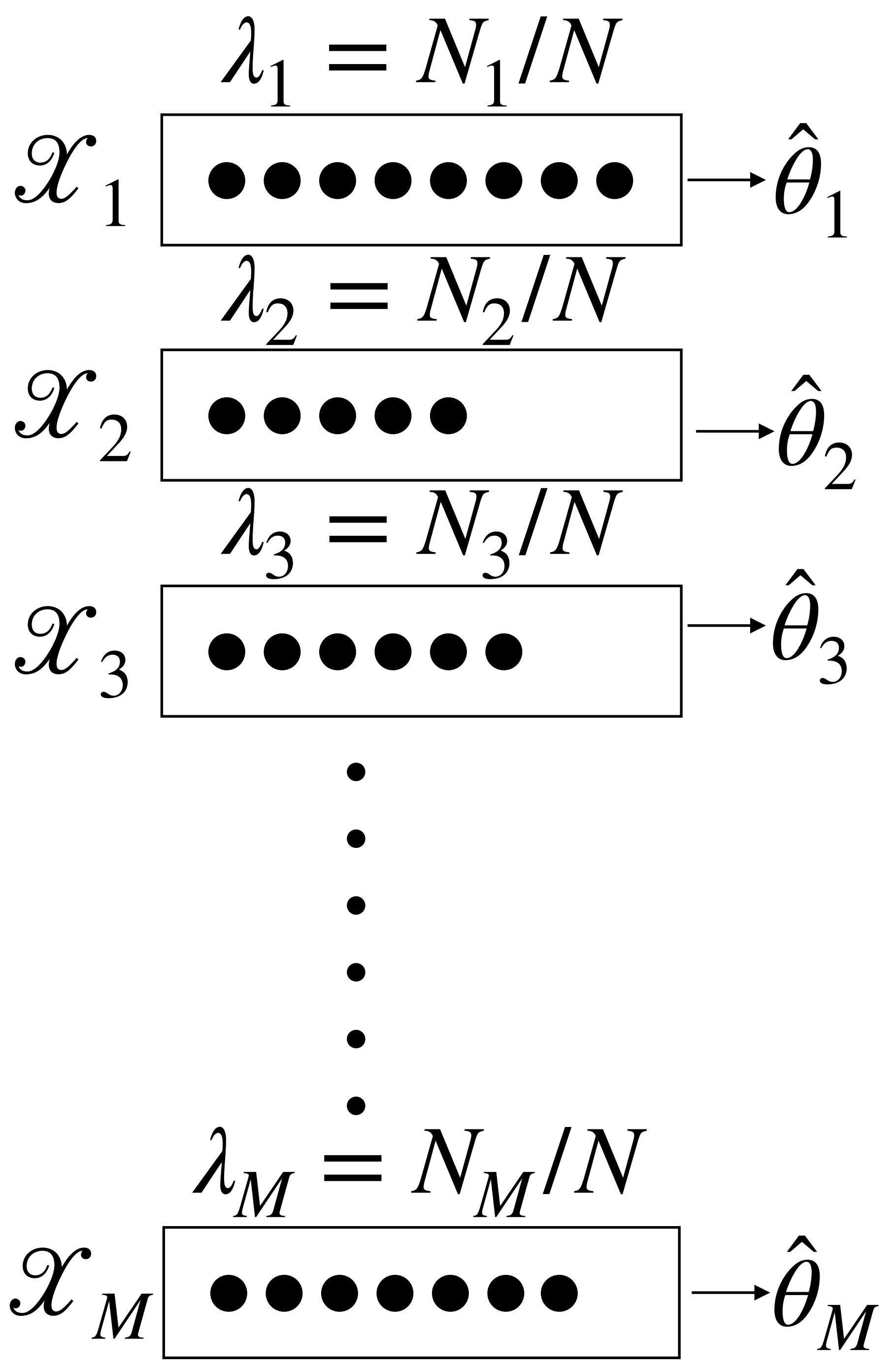
- Aggregation for real valued parameters:

$$\bar{\theta} = \sum_{m=1}^{M} \lambda_m \hat{\theta}_m$$

- Under GMM:
    - Parameter space is formed by discrete distributions with K support points.
    - Let $\bar{G} = \sum \lambda_m \hat{G}_m$
    - **Average mixture:** $\boxed{\phi(x; \bar{G}) = \sum \lambda_m \phi(x; \hat{G}_m)}$ Good estimate for true mixture

Split-and-conquer under Gaussian mixtures

Local datasets Local estimates



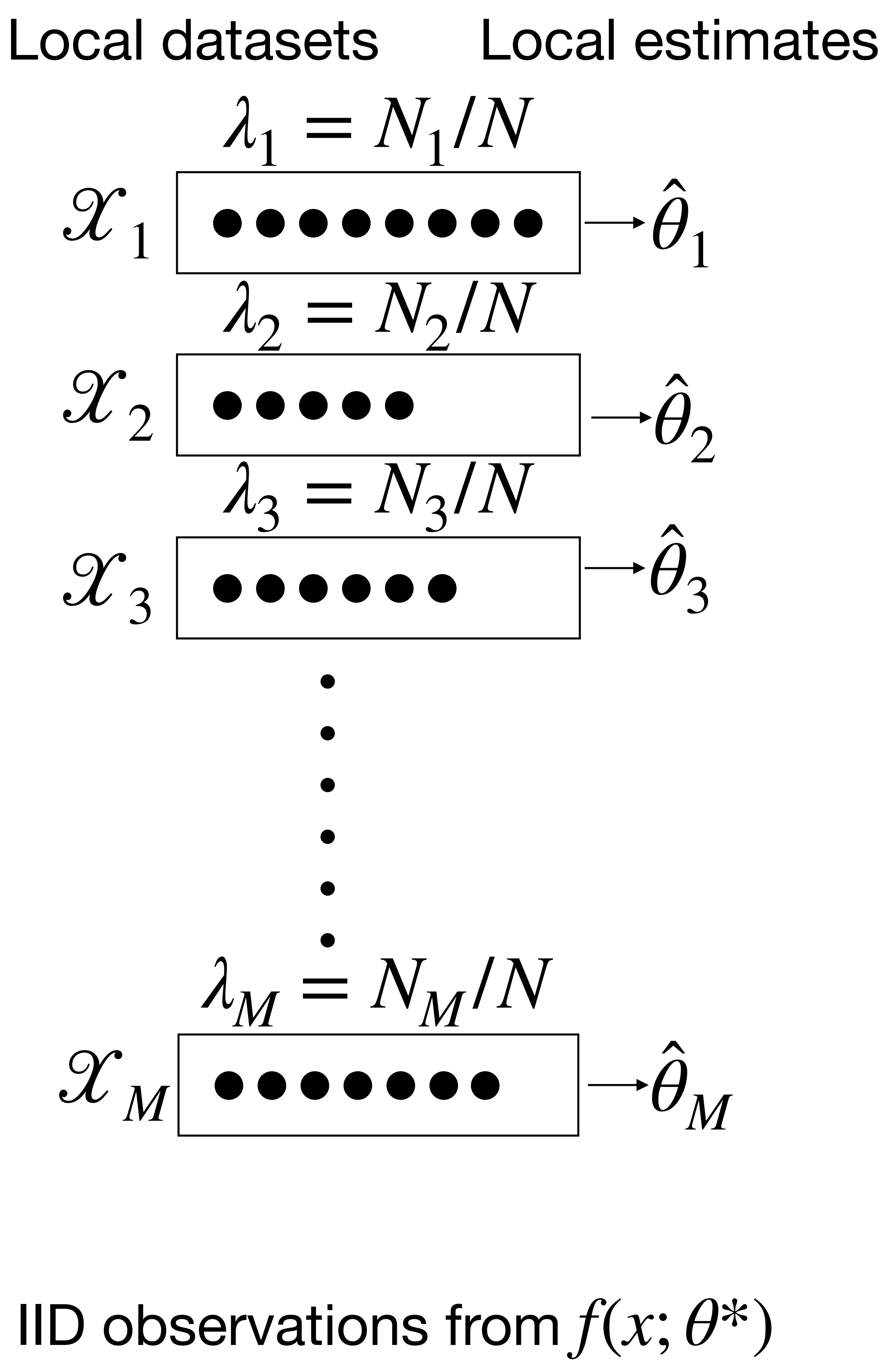
IID observations from $f(x; \theta^*)$

- Aggregation for real valued parameters:

$$\bar{\theta} = \sum_{m=1}^{M} \lambda_m \hat{\theta}_m$$

- Under GMM:
  - Parameter space is formed by discrete distributions with K support points.
  - Let $\boxed{\bar{G} = \sum \lambda_m \hat{G}_m}$ Unsatisfactory for revealing latent structure
  - **Average mixture:** $\boxed{\phi(x; \bar{G}) = \sum \lambda_m \phi(x; \hat{G}_m)}$ Good estimate for true mixture

Split-and-conquer under Gaussian mixtures



- Aggregation for real valued parameters:

$$\bar{\theta} = \sum_{m=1}^{M} \lambda_m \hat{\theta}_m$$

- Under GMM:
  - Parameter space is formed by discrete distributions with K support points.
  - Let $\boxed{\bar{G} = \sum \lambda_m \hat{G}_m}$ Unsatisfactory for revealing latent structure
  - **Average mixture:** $\boxed{\phi(x; \bar{G}) = \sum \lambda_m \phi(x; \hat{G}_m)}$ Good estimate for true mixture
- **Research problem**: aggregate local estimates under GMM

Two potential aggregation approaches

Let $\rho(\cdot\,,\cdot)$ be a divergence function that measures the similarity between two distributions

Two potential aggregation approaches

Let $\rho(\cdot\,,\cdot)$ be a divergence function that measures the similarity between two distributions

- **Barycentre**: "average" of mixing distributions

$$\bar{G}^C = \text{arginf}_{G \in \mathbb{G}_K} \sum_m \lambda_m \rho(\hat{G}_m, G)$$

(analogy of $\bar{x}_{1:n} = \text{argmin}_x \sum_{i=1}^{n} \|x_i - x\|^2$, median$(x_{1:n}) = \text{argmin}_x \sum_{i=1}^{n} |x_i - x|$ in Euclidean space)

Two potential aggregation approaches

Let $\rho(\cdot\,,\cdot)$ be a divergence function that measures the similarity between two distributions

- **Barycentre**: "average" of mixing distributions

$$\bar{G}^C = \mathrm{arginf}_{G \in \mathbb{G}_K} \sum_m \lambda_m \rho(\hat{G}_m, G)$$

(analogy of $\bar{x}_{1:n} = \mathrm{argmin}_x \sum_{i=1}^{n} \|x_i - x\|^2$, median$(x_{1:n}) = \mathrm{argmin}_x \sum_{i=1}^{n} |x_i - x|$ in Euclidean space)

- **Reduction**: approximate average mixture by an order K mixture

$$\bar{G}^R = \mathrm{arginf}_{G \in \mathbb{G}_K} \rho(\bar{G}, G)$$

Connection of two aggregation approaches

- When

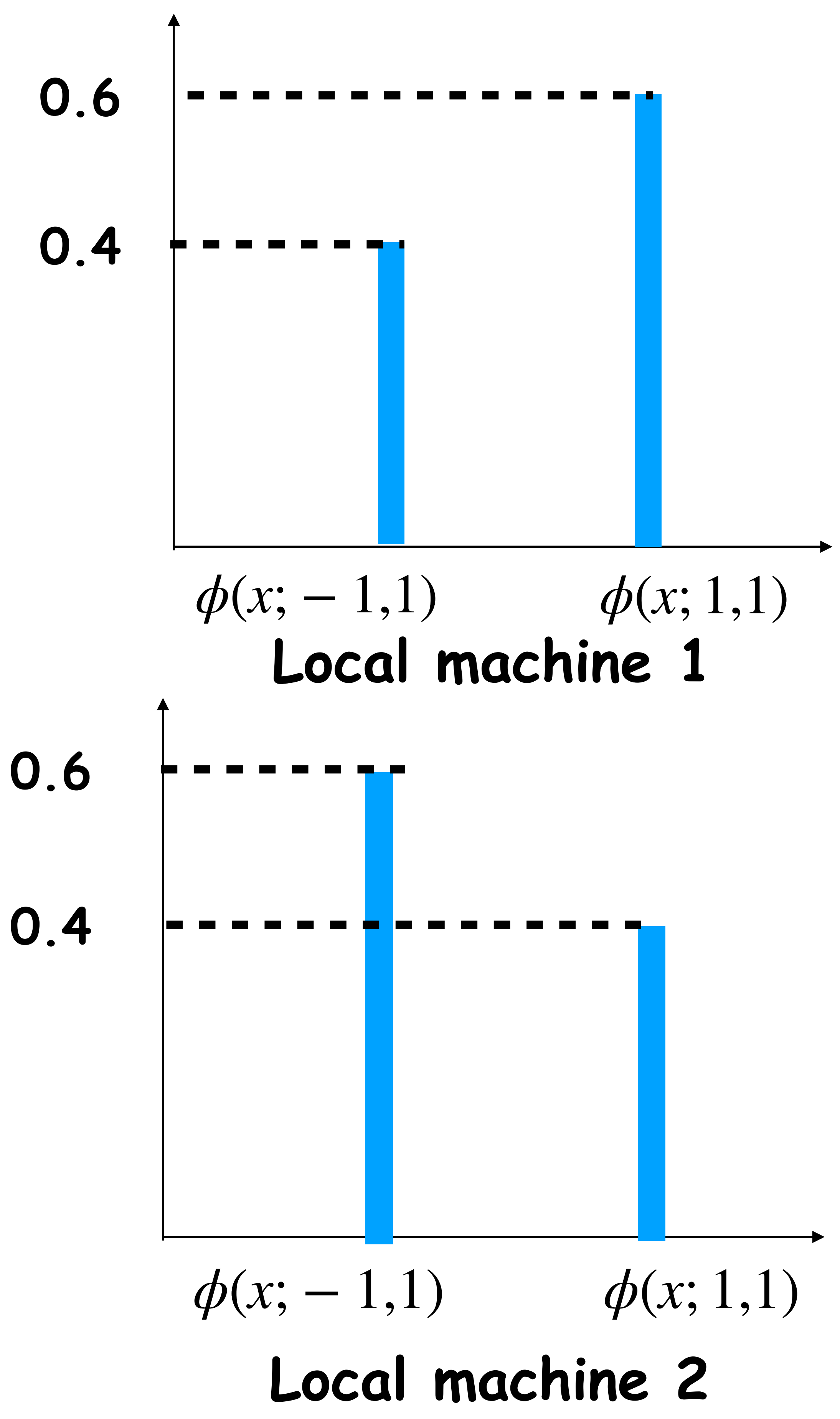
$$\rho(G_1, G_2) = D_{\mathrm{KL}}(\Phi(\,\cdot\,; G_1) \| \Phi(\,\cdot\,; G_2))$$
$$= \int \phi(x; G_1) \log \frac{\phi(x; G_1)}{\phi(x; G_2)}\, dx$$

then

$$\bar{G}^C = \bar{G}^R$$

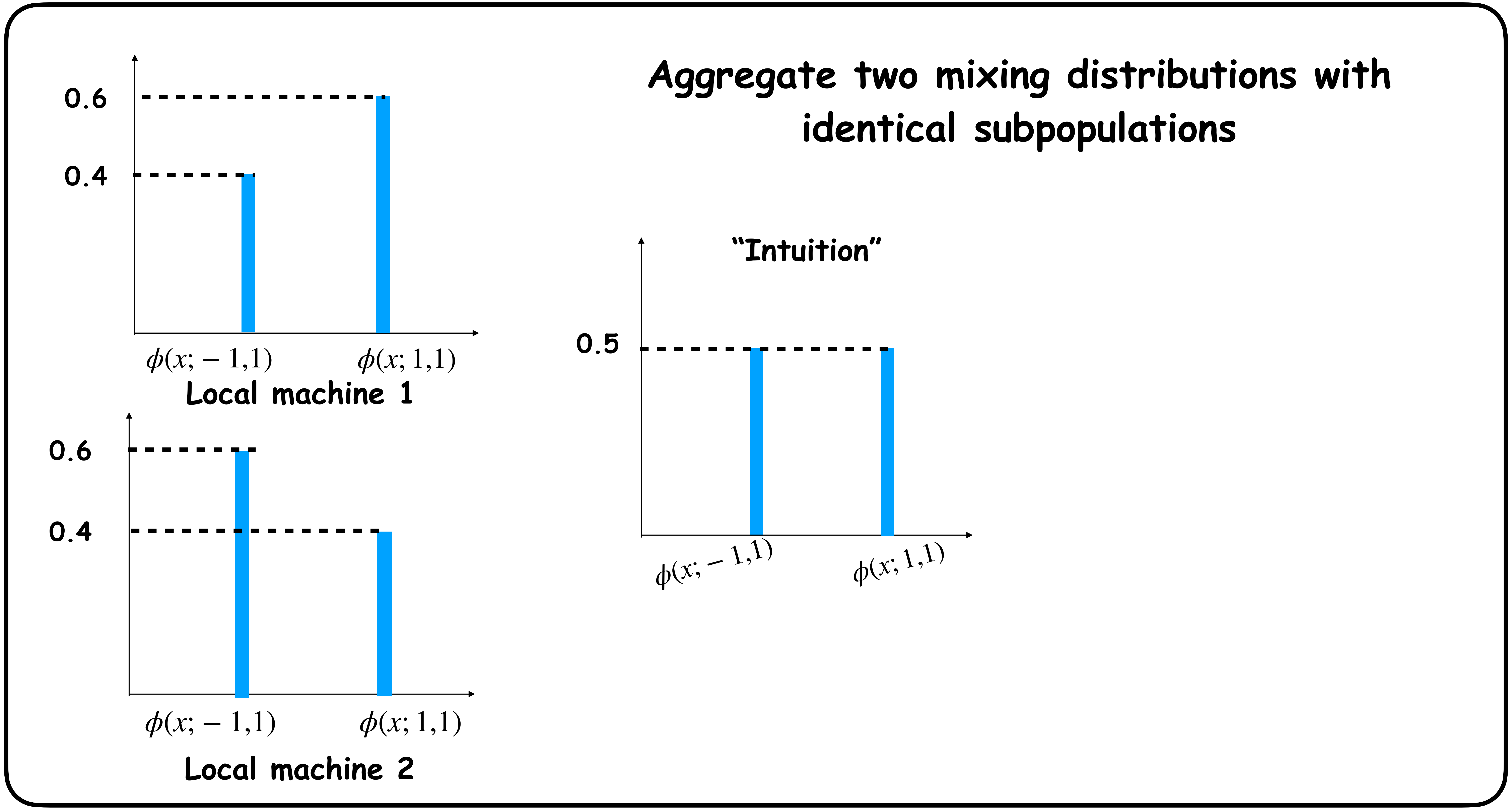
- However, exact solution is computationally intractable

Barycenter approach may not be ideal

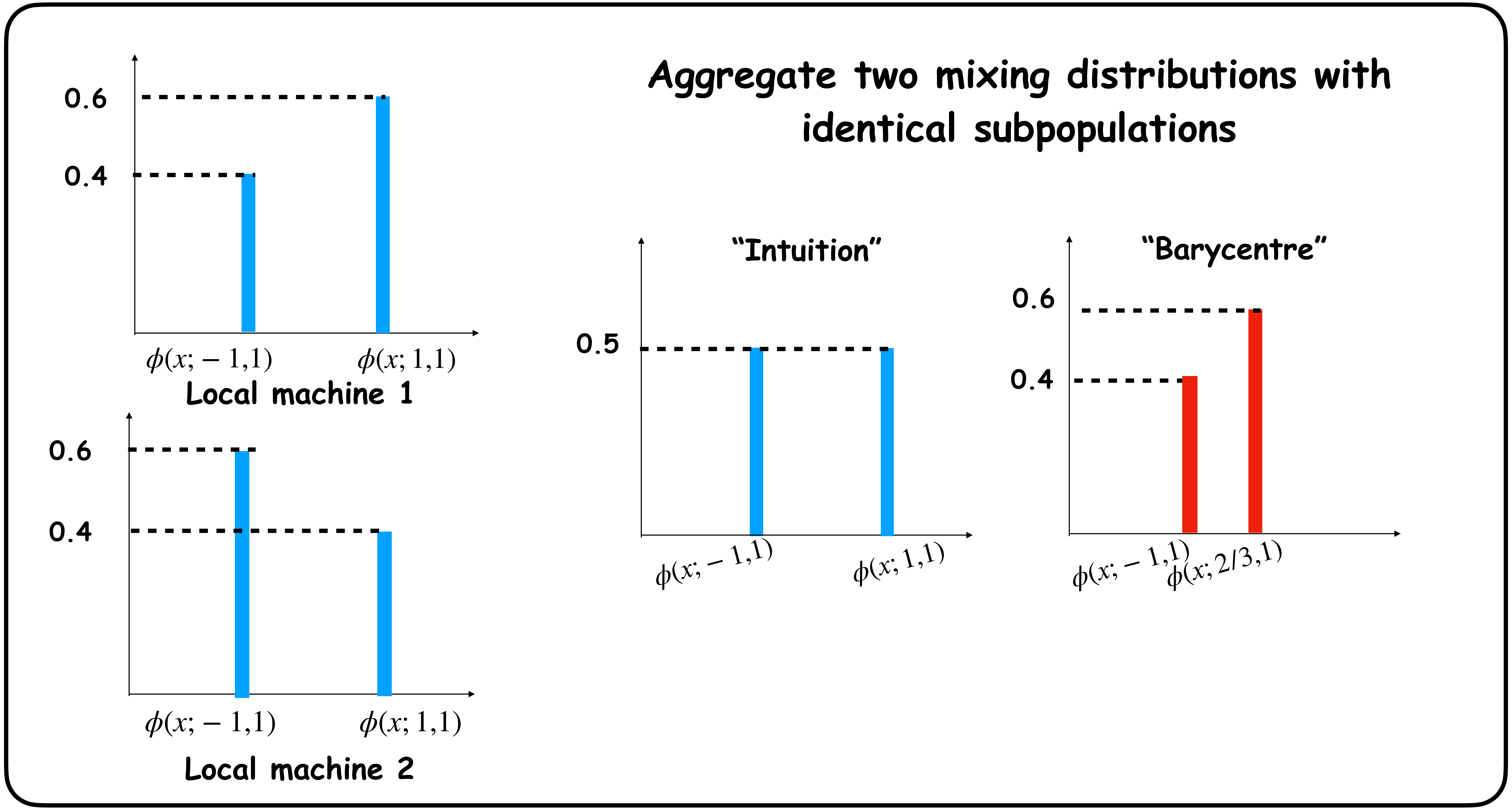


Aggregate two mixing distributions with identical subpopulations

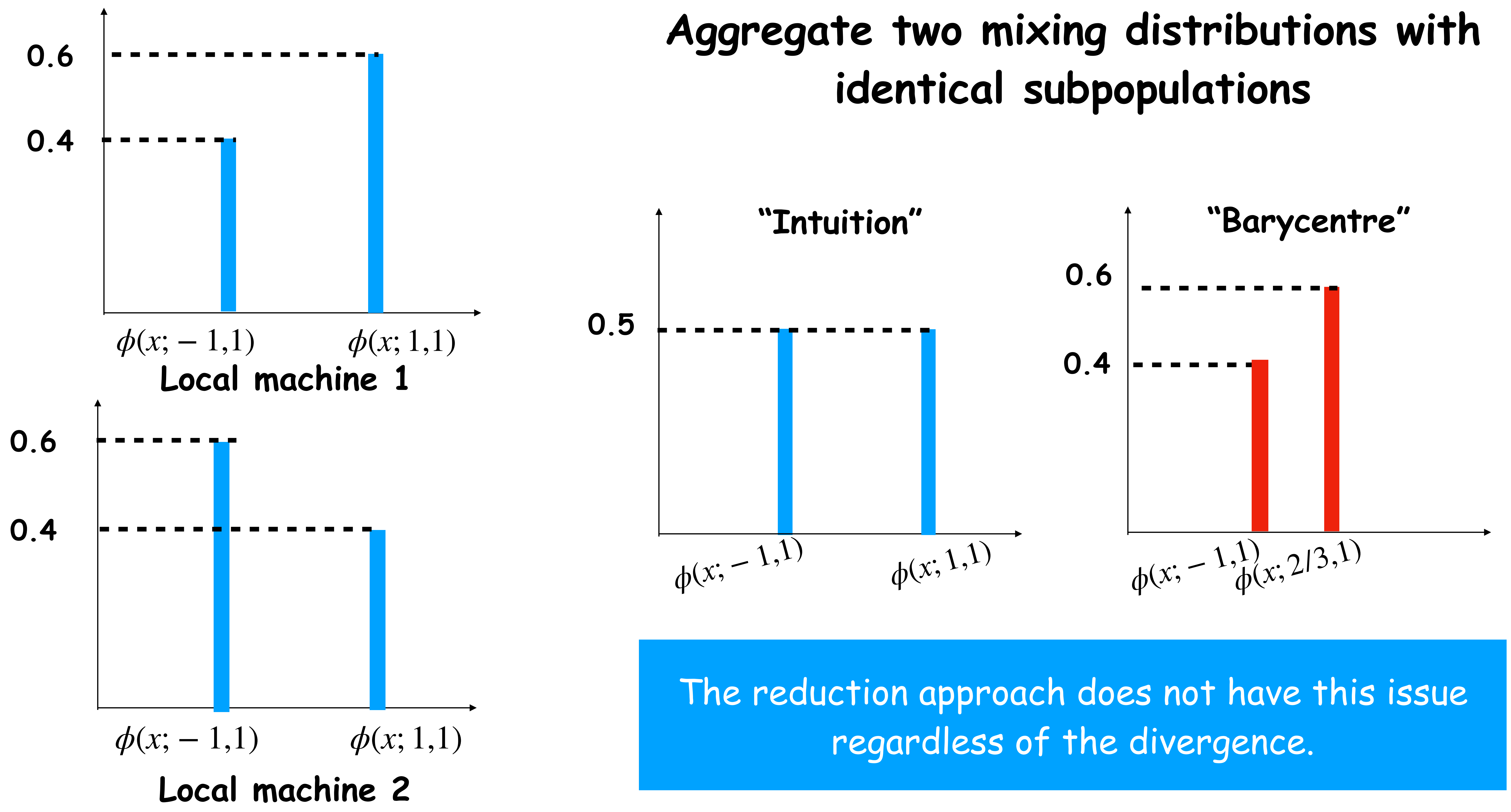
Barycenter approach may not be ideal



Barycenter approach may not be ideal



Barycenter approach may not be ideal



The reduction approach does not have this issue regardless of the divergence.

Which divergence?

- We propose to aggregate via the **reduction** approach

$$\bar{G}^R = \text{arginf}_{G \in \mathbb{G}_K} \rho(\bar{G}, G) \,.$$

Which divergence?

- We propose to aggregate via the **reduction** approach

$$\bar{G}^R = \text{arginf}_{G \in \mathbb{G}_K} \rho(\bar{G}, G)\,.$$

- Which divergence $\rho(\,\cdot\,,\,\cdot\,)$ should we pick?

Which divergence?

- We propose to aggregate via the **reduction** approach

$$\bar{G}^R = \text{arginf}_{G \in \mathbb{G}_K} \rho(\bar{G}, G) \,.$$

- Which divergence $\rho(\,\cdot\,,\,\cdot\,)$ should we pick?
  - Key observation:
    - divergence is hard to compute between mixtures
    - divergence is easy to compute between Gaussians

Which divergence?

- We propose to aggregate via the **reduction** approach

$$\bar{G}^R = \text{arginf}_{G \in \mathbb{G}_K} \rho(\bar{G}, G)\,.$$

- Which divergence $\rho(\,\cdot\,,\,\cdot\,)$ should we pick?
  - Key observation:
    - divergence is hard to compute between mixtures
    - divergence is easy to compute between Gaussians
- The divergence we used: composite transportation divergence
  - A byproduct of optimal transport

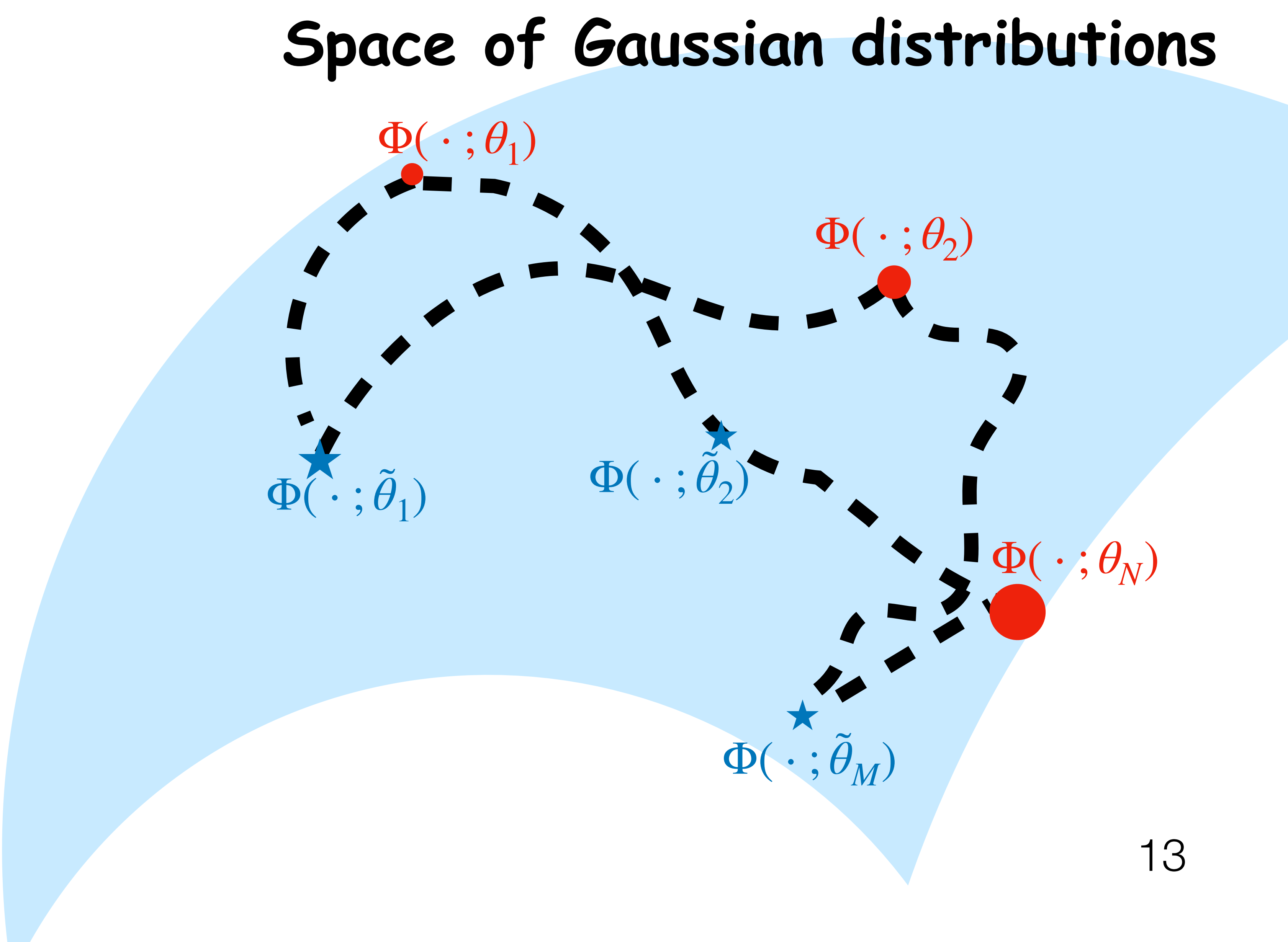
Proposed method

Composite transportation divergence and proposed method

Composite transportation divergence between two Gaussian mixtures (Chen et al. 2019)

Let $\Phi(x; G) = \sum_{n=1}^{N} w_n \Phi(x; \theta_n)$ and $\Phi(x; \tilde{G}) = \sum_{m=1}^{M} \tilde{w}_m \Phi(x; \tilde{\theta}_m)$ and $c(\cdot, \cdot) : \mathscr{F} \times \mathscr{F} \to \mathbb{R}_+$ be the cost function which is a divergence on $\mathscr{F}$. The Composite transportation divergence between $\Phi(x; G)$ and $\Phi(x; \tilde{G})$ is defined to be

$$\mathscr{T}_c(\Phi(\cdot\,; G), \Phi(\cdot\,; \tilde{G})) = \min \left\{ \sum_{n,m} \pi_{nm} c(\Phi(\cdot\,; \theta_n), \Phi(\cdot\,; \tilde{\theta}_m)) : \sum_{m} \pi_{nm} = w_n, \sum_{n} \pi_{nm} = \tilde{w}_m \right\}$$



Composite transportation divergence and proposed method

Composite transportation divergence between two Gaussian mixtures (Chen et al. 2019)

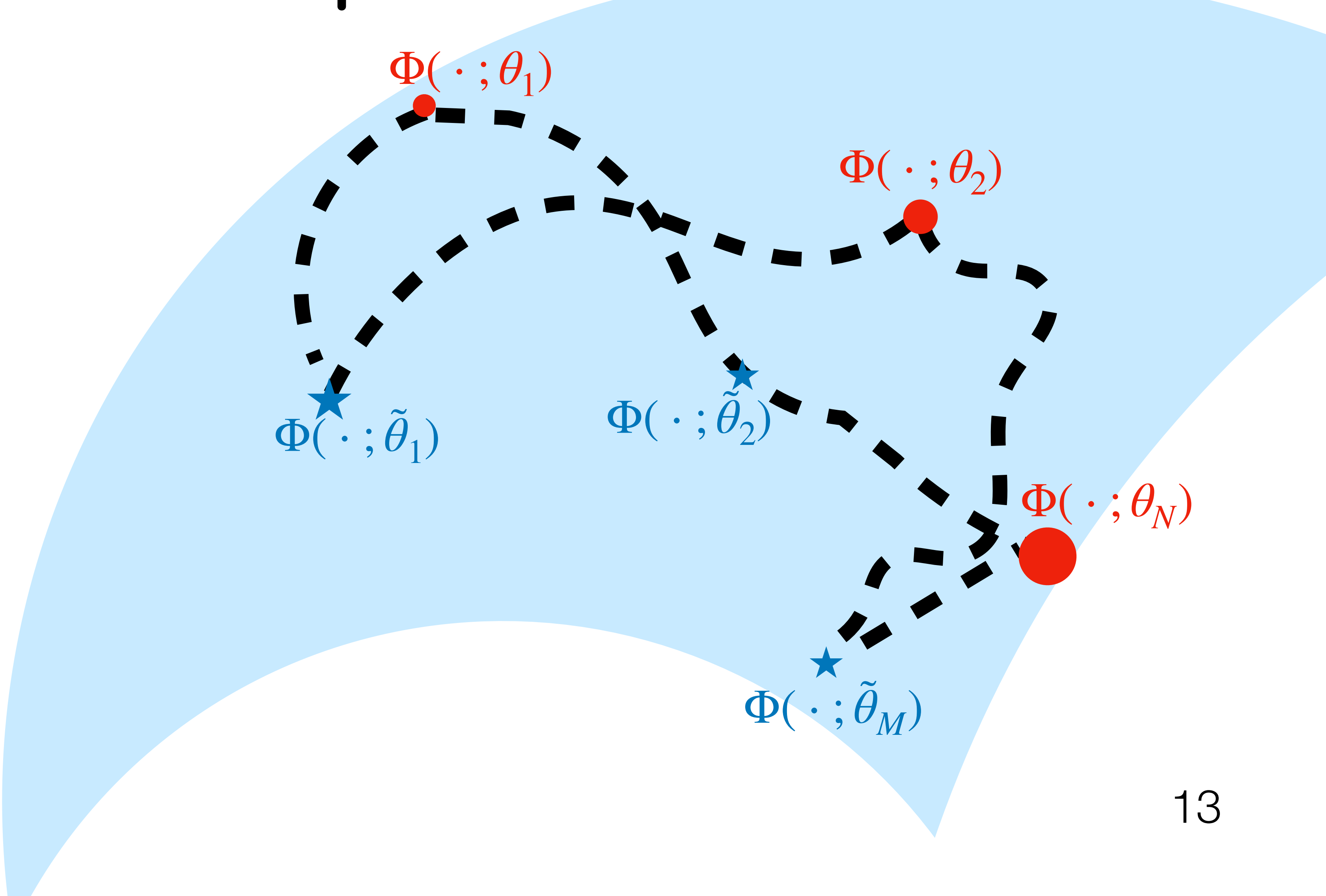
Let $\Phi(x;G) = \sum_{n=1}^{N} w_n \Phi(x;\theta_n)$ and $\Phi(x;\tilde{G}) = \sum_{m=1}^{M} \tilde{w}_m \Phi(x;\tilde{\theta}_m)$ and $c(\cdot,\cdot) : \mathcal{F} \times \mathcal{F} \to \mathbb{R}_+$ be the cost function which is a divergence on $\mathcal{F}$. The Composite transportation divergence between $\Phi(x;G)$ and $\Phi(x;\tilde{G})$ is defined to be

$$\mathcal{T}_c(\Phi(\cdot;G), \Phi(\cdot;\tilde{G})) = \min \left\{ \sum_{n,m} \pi_{nm} c(\Phi(\cdot;\theta_n), \Phi(\cdot;\tilde{\theta}_m)) : \sum_m \pi_{nm} = w_n, \sum_n \pi_{nm} = \tilde{w}_m \right\}$$

Our proposed aggregated estimator is

$$\bar{G}^R = \text{arginf}_{G \in \mathbb{G}_K} \mathcal{T}_c(\Phi(\cdot;\bar{G}), \Phi(\cdot;G)) := \text{arginf}_{G \in \mathbb{G}_K} \mathcal{T}_c(G)$$

Space of Gaussian distributions



Composite transportation divergence and proposed method

Composite transportation divergence between two Gaussian mixtures (Chen et al. 2019)

Let $\Phi(x; G) = \sum_{n=1}^{N} w_n \Phi(x; \theta_n)$ and $\Phi(x; \tilde{G}) = \sum_{m=1}^{M} \tilde{w}_m \Phi(x; \tilde{\theta}_m)$ and $c(\cdot, \cdot) : \mathscr{F} \times \mathscr{F} \to \mathbb{R}_+$ be the cost function which is a divergence on $\mathscr{F}$. The Composite transportation divergence between $\Phi(x; G)$ and $\Phi(x; \tilde{G})$ is defined to be

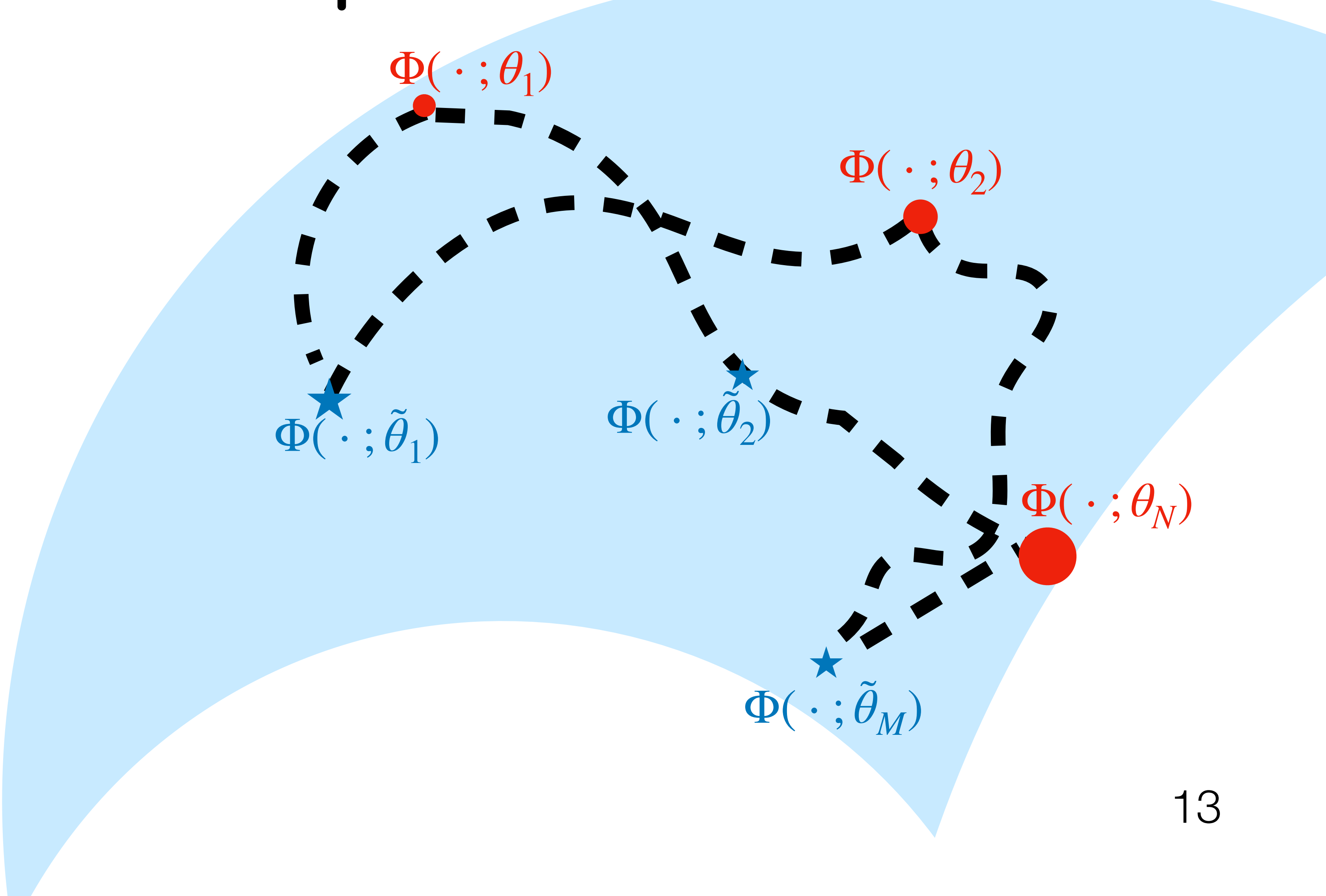
$$\mathscr{T}_c(\Phi(\cdot\,; G), \Phi(\cdot\,; \tilde{G})) = \min \left\{ \sum_{n,m} \pi_{nm} c(\Phi(\cdot\,; \theta_n), \Phi(\cdot\,; \tilde{\theta}_m)) : \sum_m \pi_{nm} = w_n, \sum_n \pi_{nm} = \tilde{w}_m \right\}$$

Our proposed aggregated estimator is

$$\bar{G}^R = \text{arginf}_{G \in \mathbb{G}_K} \mathscr{T}_c(\Phi(\cdot\,; \bar{G}), \Phi(\cdot\,; G)) := \text{arginf}_{G \in \mathbb{G}_K} \mathscr{T}_c(G)$$

How to compute the aggregated estimator numerically?

Space of Gaussian distributions



A glance at the numerical computation

Our optimization problem

$$\bar{G}^R = \text{arginf}_{G \in \mathbb{G}_K} \mathcal{T}_c(G)$$

$$\mathcal{T}_c(G) = \min \left\{ \sum_{n,m} \pi_{nm} c(\Phi(\,\cdot\,; \theta_n), \Phi(\,\cdot\,; \tilde{\theta}_m)) : \sum_m \pi_{nm} = w_n, \sum_n \pi_{nm} = \tilde{w}_m \right\}$$

A glance at the numerical computation

Our optimization problem

$$\bar{G}^R = \mathrm{arginf}_{G \in \mathbb{G}_K} \mathcal{T}_c(G)$$

$$\mathcal{T}_c(G) = \min \left\{ \sum_{n,m} \pi_{nm} c(\Phi(\,\cdot\,;\theta_n), \Phi(\,\cdot\,;\tilde{\theta}_m)) : \sum_m \pi_{nm} = w_n, \sum_n \pi_{nm} = \tilde{w}_m \right\}$$

- Bilevel optimization: the **objective function** itself involves another optimization problem
- We find
    - Step I: A simplified equivalent objective with a closed form
    - Step 2: an MM algorithm to minimize the simplified objective

Numerical algorithm

Step I: Simplified Optimization Problem

Given $\bar{G}$, for $G \in \mathbb{G}_K$, let

$$\mathcal{J}_c(G) = \min_{\pi} \left\{ \sum_{n,m} \pi_{nm} c(\Phi(\,\cdot\,;\bar{\theta}_n), \Phi(\,\cdot\,;\theta_m)) : \sum_{m=1}^{M} \pi_{nm} = \bar{w}_n, \cancel{\sum_{n=1}^{N} \pi_{nm} = w_m} \right\},$$

Numerical algorithm

Step I: Simplified Optimization Problem

$$\mathcal{J}_c(G) = \sum_{n,m} \pi^*_{nm}(G) c(\Phi(\,\cdot\,; \bar{\theta}_n), \Phi(\,\cdot\,; \theta_m)) \text{ where}$$

$$\pi^*_{nm}(G) = \begin{cases} \bar{w}_n & m = \operatorname{argmin}_{m'} c(\bar{\Phi}_n, \Phi_{m'}) \\ 0 & \text{otherwise}. \end{cases}$$

Closed form

Given $\bar{G}$, for $G \in \mathbb{G}_K$, let

$$\mathcal{J}_c(G) = \min_{\pi} \left\{ \sum_{n,m} \pi_{nm} c(\Phi(\,\cdot\,; \bar{\theta}_n), \Phi(\,\cdot\,; \theta_m)) : \sum_{m=1}^{M} \pi_{nm} = \bar{w}_n, \cancel{\sum_{n=1}^{N} \pi_{nm} = w_m} \right\},$$

Numerical algorithm

Step I: Simplified Optimization Problem

$$\mathcal{J}_c(G) = \sum_{n,m} \pi^*_{nm}(G) c(\Phi(\,\cdot\,; \bar{\theta}_n), \Phi(\,\cdot\,; \theta_m)) \text{ where}$$

$$\pi^*_{nm}(G) = \begin{cases} \bar{w}_n & m = \text{argmin}_{m'} c(\bar{\Phi}_n, \Phi_{m'}) \\ 0 & \text{otherwise} . \end{cases}$$

Closed form

Given $\bar{G}$, for $G \in \mathbb{G}_K$, let

$$\mathcal{J}_c(G) = \min_{\pi} \left\{ \sum_{n,m} \pi_{nm} c(\Phi(\,\cdot\,; \bar{\theta}_n), \Phi(\,\cdot\,; \theta_m)) : \sum_{m=1}^{M} \pi_{nm} = \bar{w}_n, \cancel{\sum_{n=1}^{N} \pi_{nm} = w_m} \right\},$$

We have

$$\inf\{\mathcal{T}_c(G) : G \in \mathbb{G}_K\} = \inf\{\mathcal{J}_c(G) : G \in \mathbb{G}_K\}$$

with mixing distribution

$$w_m = \sum_{n=1}^{N} \pi^*_{nm}(\bar{G}^R)$$

Numerical algorithm

Step I: Simplified Optimization Problem

$$\mathcal{J}_c(G) = \sum_{n,m} \pi^*_{nm}(G) c(\Phi(\,\cdot\,;\bar{\theta}_n), \Phi(\,\cdot\,;\theta_m)) \text{ where}$$

$$\pi^*_{nm}(G) = \begin{cases} \bar{w}_n & m = \text{argmin}_{m'} c(\bar{\Phi}_n, \Phi_{m'}) \\ 0 & \text{otherwise}\,. \end{cases}$$

Closed form

Given $\bar{G}$, for $G \in \mathbb{G}_K$, let

$$\mathcal{J}_c(G) = \min_{\pi} \left\{ \sum_{n,m} \pi_{nm} c(\Phi(\,\cdot\,;\bar{\theta}_n), \Phi(\,\cdot\,;\theta_m)) : \sum_{m=1}^{M} \pi_{nm} = \bar{w}_n, \cancel{\sum_{n=1}^{N} \pi_{nm} = w_m} \right\},$$

We have

$$\inf\{\mathcal{T}_c(G) : G \in \mathbb{G}_K\} = \inf\{\mathcal{J}_c(G) : G \in \mathbb{G}_K\}$$

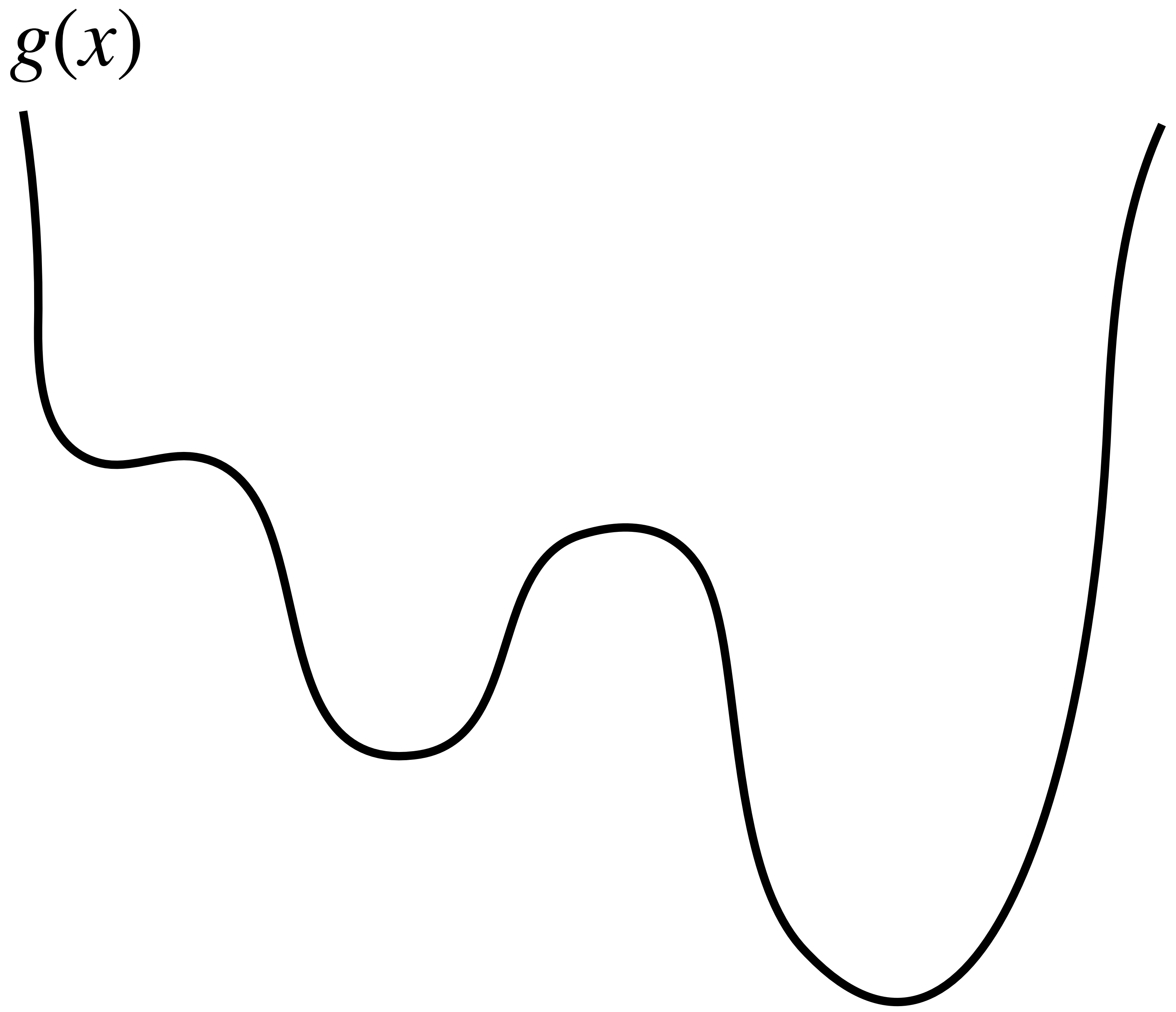
with mixing distribution

$$w_m = \sum_{n=1}^{N} \pi^*_{nm}(\bar{G}^R)$$

- **Pros**
  - The subpopulation parameters and mixing weights can be updated separately
  - Allows for an efficient MM algorithm (update G and $\pi^*(G)$ iteratively)

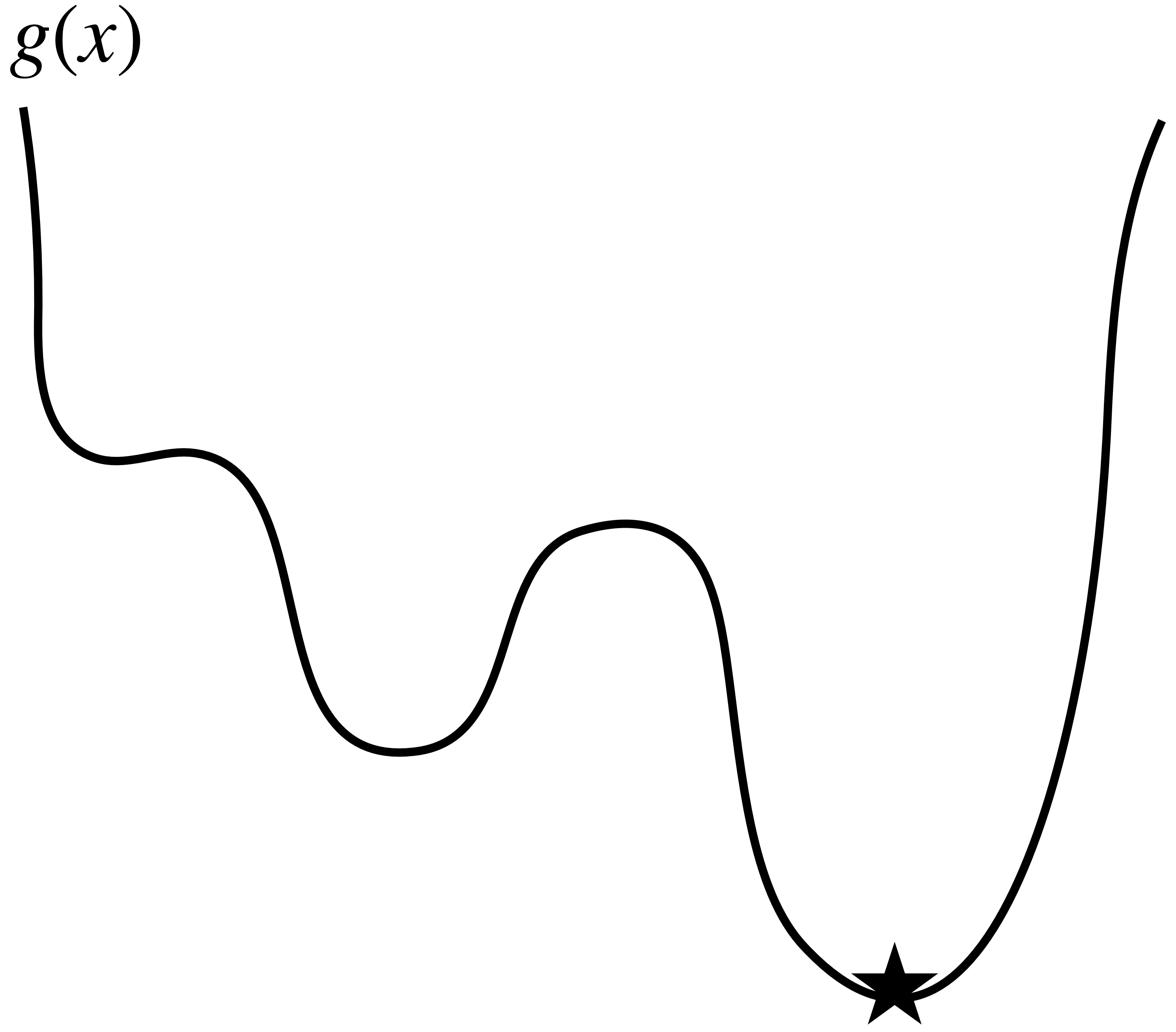
Numerical algorithm

Step II: MM Algorithm (iteratively update transportation plan and the target location)



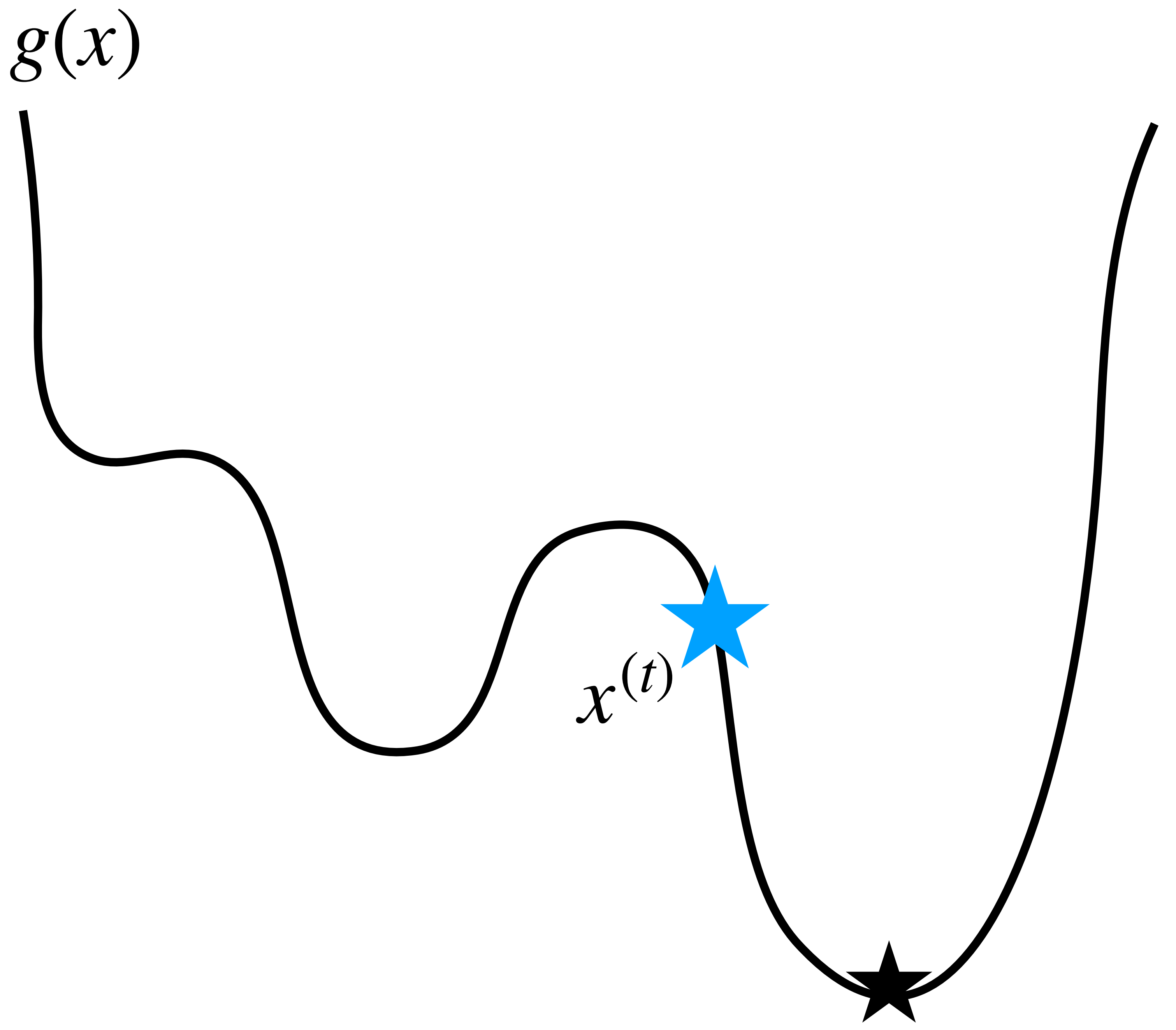
Numerical algorithm

Step II: MM Algorithm (iteratively update transportation plan and the target location)



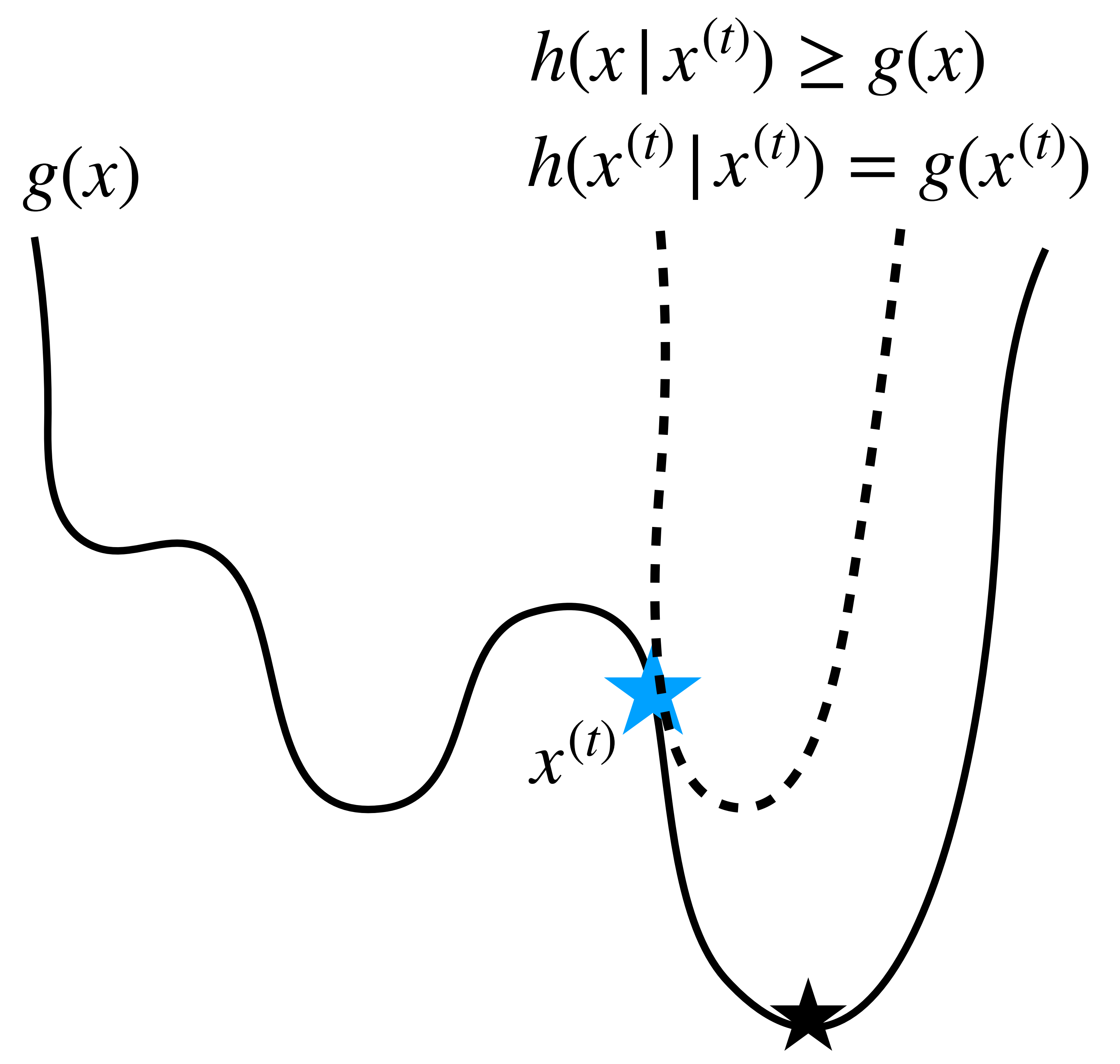
Numerical algorithm

Step II: MM Algorithm (iteratively update transportation plan and the target location)



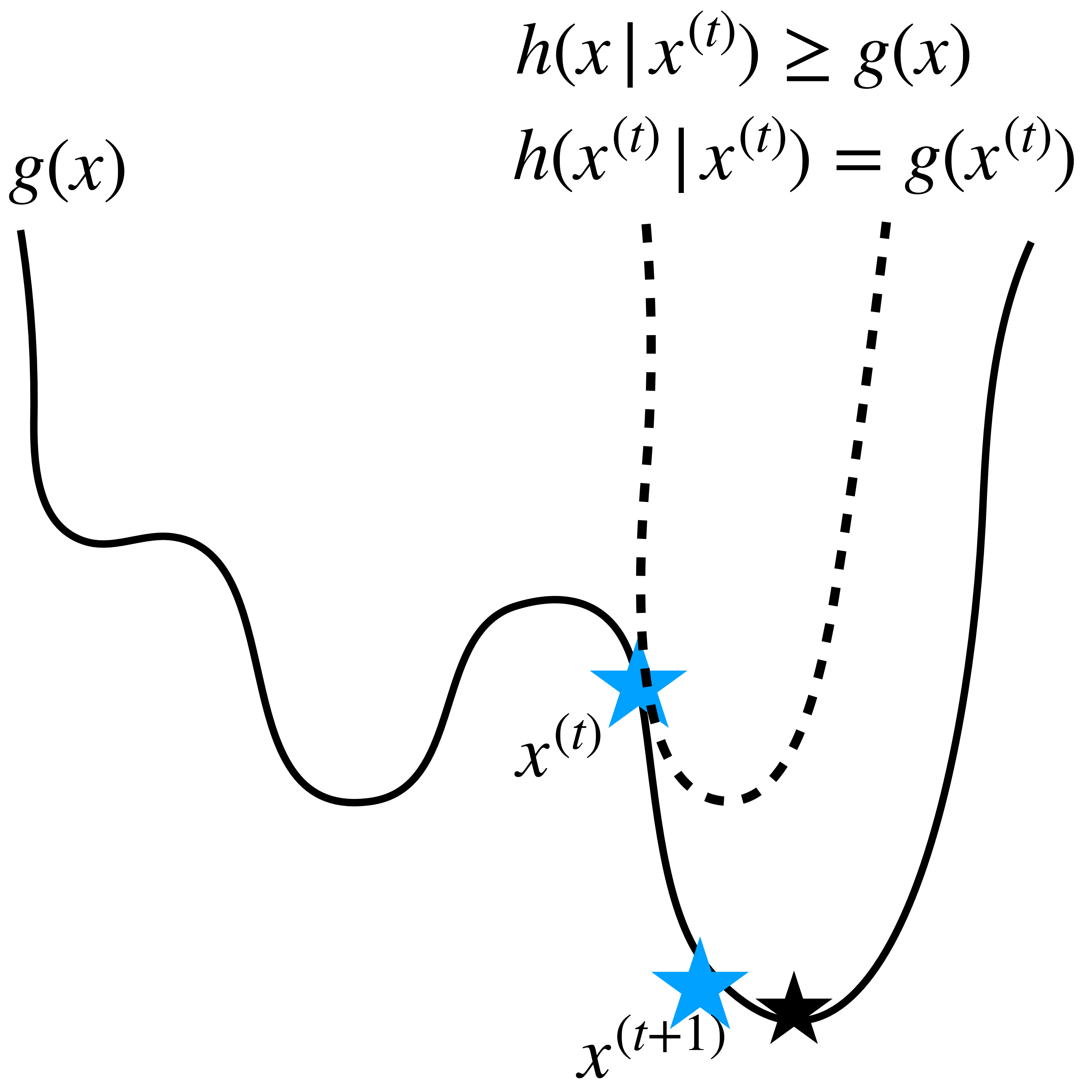
Numerical algorithm

Step II: MM Algorithm (iteratively update transportation plan and the target location)



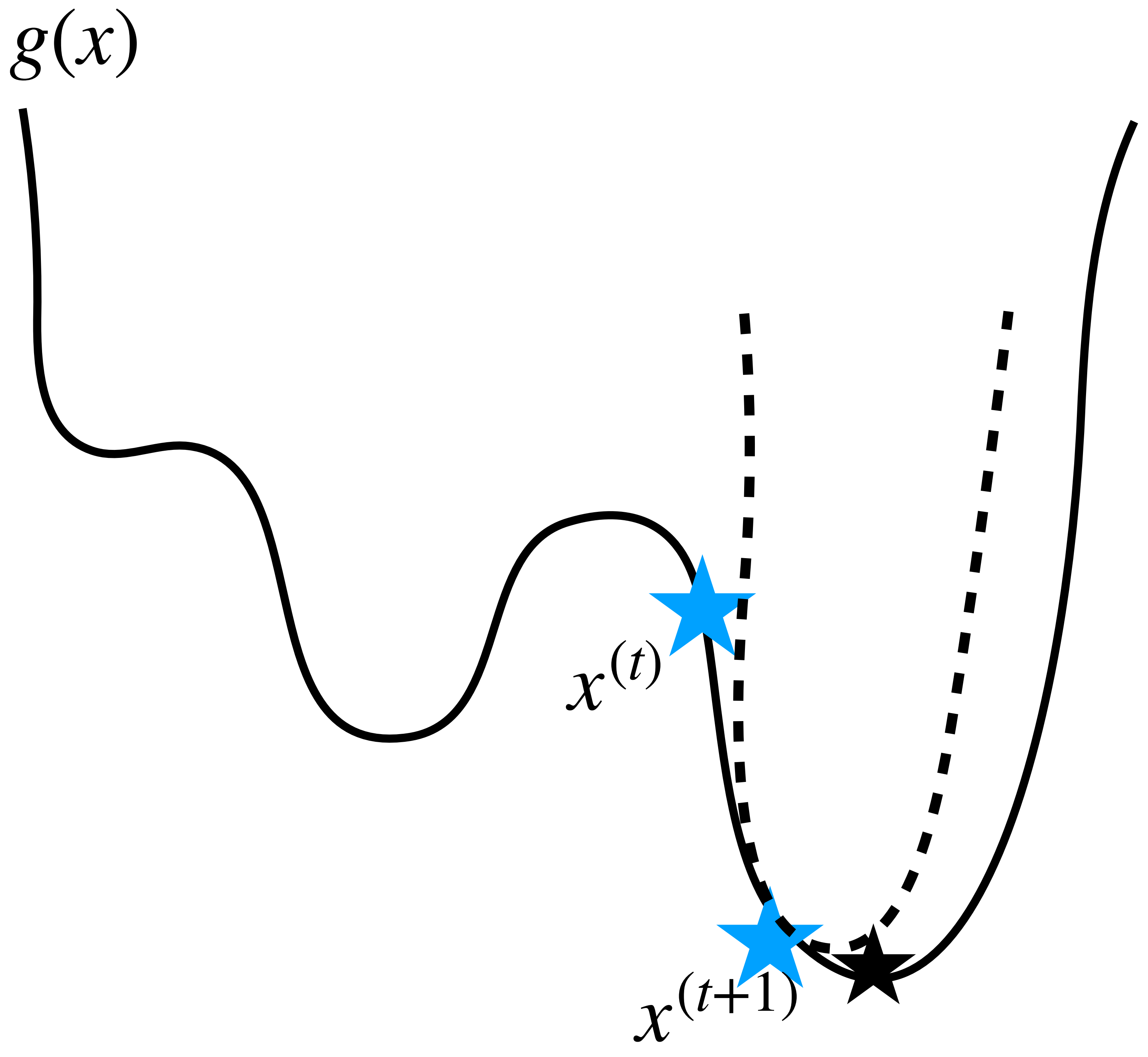
Numerical algorithm

Step II: MM Algorithm (iteratively update transportation plan and the target location)



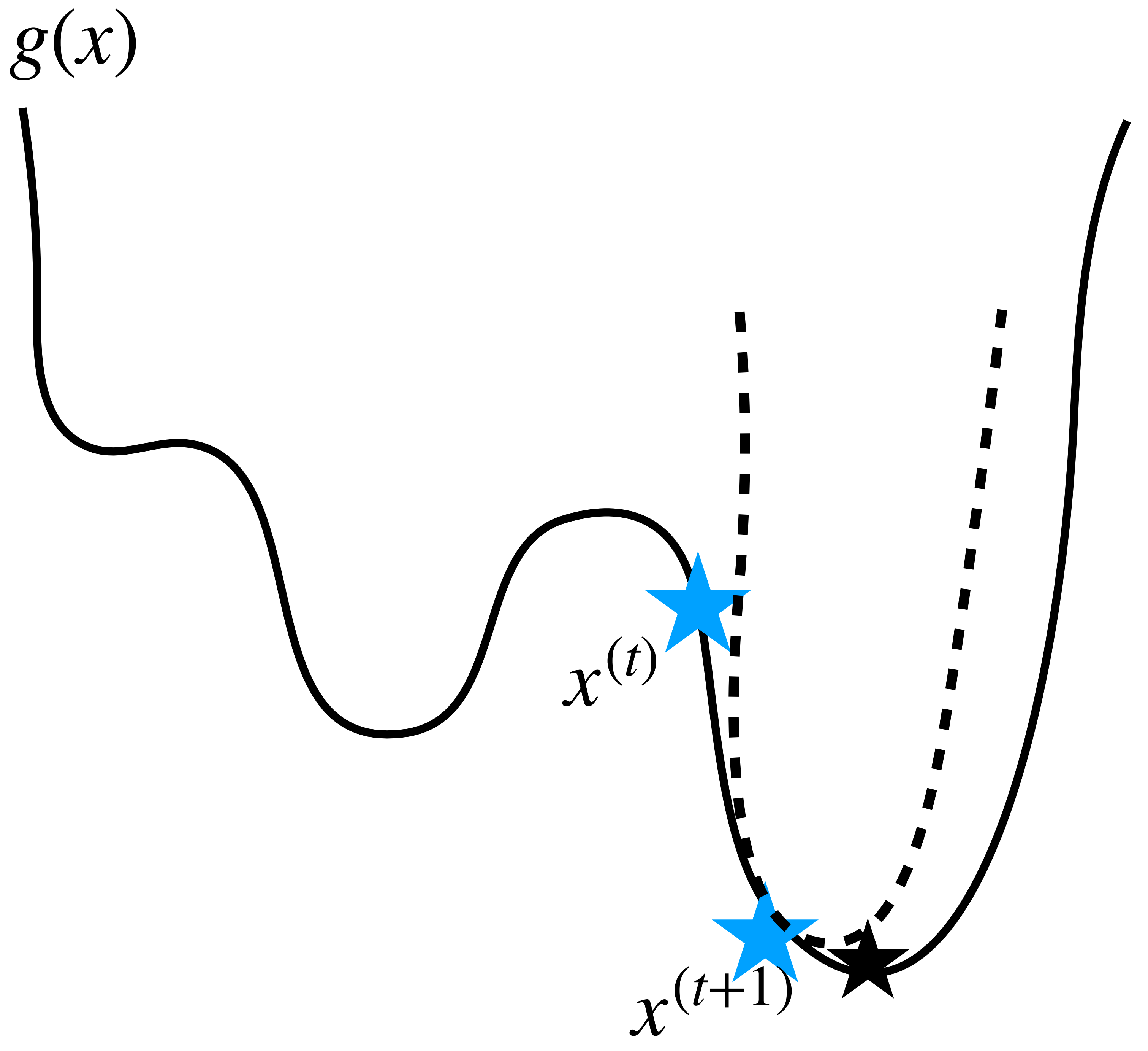
Numerical algorithm

Step II: MM Algorithm (iteratively update transportation plan and the target location)



Numerical algorithm

Step II: MM Algorithm (iteratively update transportation plan and the target location)

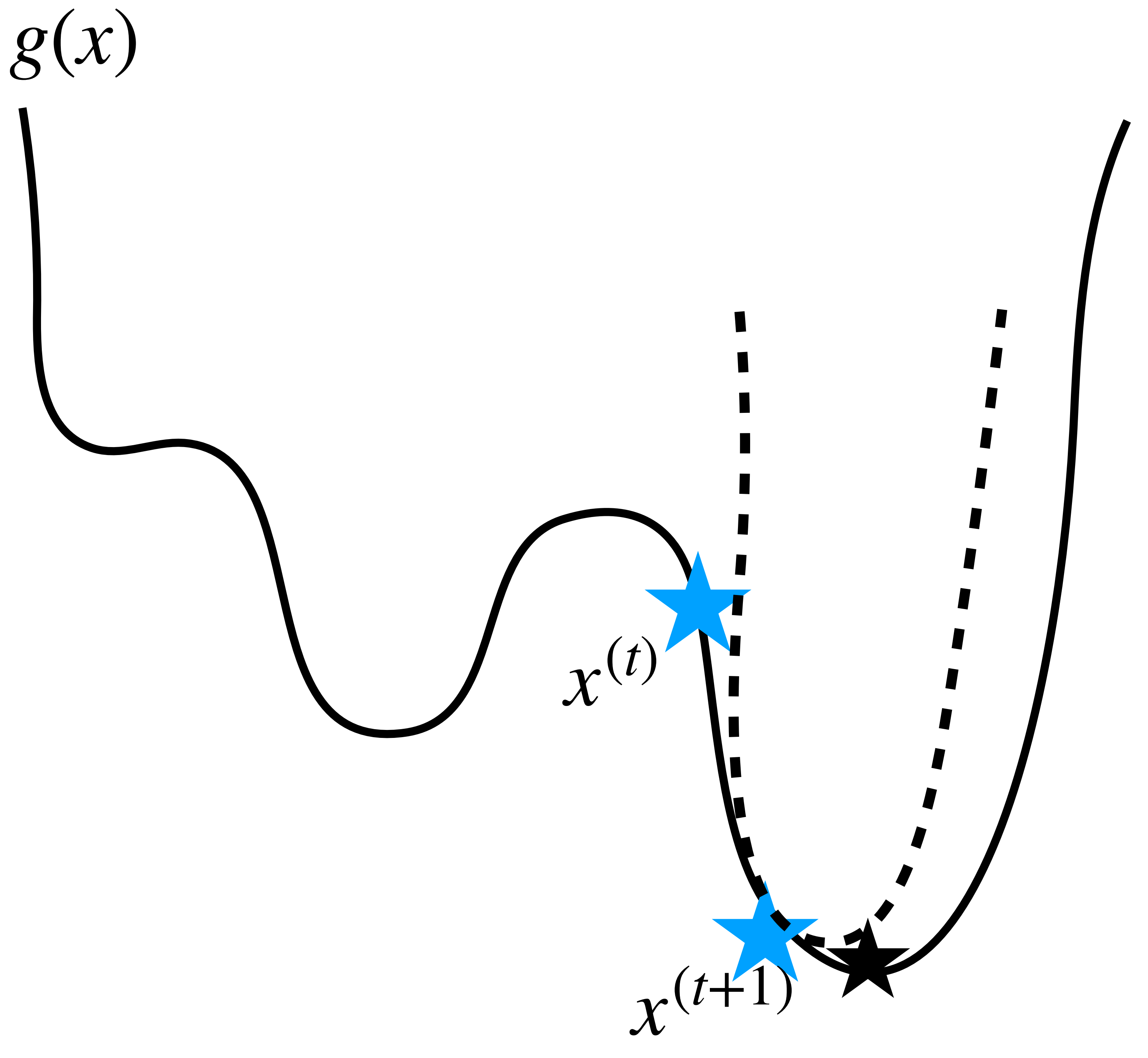


- Objective

$$\mathcal{J}_c(G) = \min \left\{ \sum_{n,m} \pi_{nm} c(\Phi(\,\cdot\,;\bar{\theta}_n), \Phi(\,\cdot\,;\theta_m)) : \sum_m \pi_{nm} = \bar{w}_n \right\}$$

Numerical algorithm

Step II: MM Algorithm (iteratively update transportation plan and the target location)



- Objective

$$\mathcal{J}_c(G) = \min \left\{ \sum_{n,m} \pi_{nm} c(\Phi(\cdot\,;\bar{\theta}_n), \Phi(\cdot\,;\theta_m)) : \sum_m \pi_{nm} = \bar{w}_n \right\}$$

- Majorization function at $G^{(t)}$

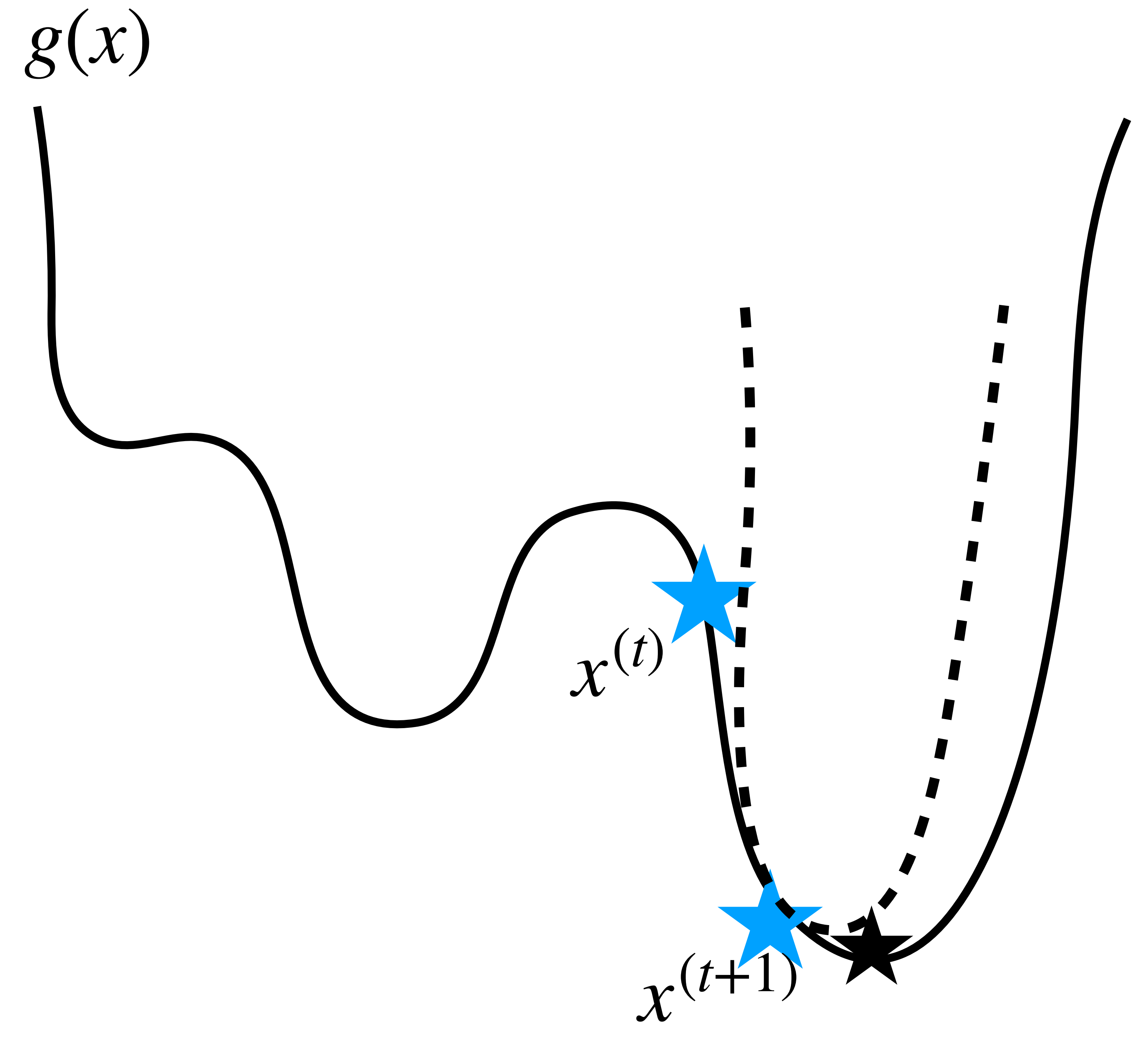
$$\mathcal{K}_c(G \,|\, G^{(t)}) = \sum_{n,m} \pi^*_{nm}(G^{(t)}) c(\Phi(\cdot\,;\bar{\theta}_n), \Phi(\cdot\,;\theta_m))$$

with

$$\pi^*_{nm}(G^{(t)}) = \begin{cases} \bar{w}_n & m = \text{argmin}_{m'} c(\Phi(\cdot\,;\bar{\theta}_n), \Phi(\cdot\,;\theta^{(t)}_{m'})) \\ 0 & \text{otherwise}\,. \end{cases}$$

Numerical algorithm

Step II: MM Algorithm (iteratively update transportation plan and the target location)



- Objective

$$\mathscr{J}_c(G) = \min \left\{ \sum_{n,m} \pi_{nm} c(\Phi(\cdot\,;\bar{\theta}_n), \Phi(\cdot\,;\theta_m)) : \sum_m \pi_{nm} = \bar{w}_n \right\}$$

- Majorization function at $G^{(t)}$

$$\mathscr{K}_c(G \mid G^{(t)}) = \sum_{n,m} \pi^*_{nm}(G^{(t)}) c(\Phi(\cdot\,;\bar{\theta}_n), \Phi(\cdot\,;\theta_m))$$

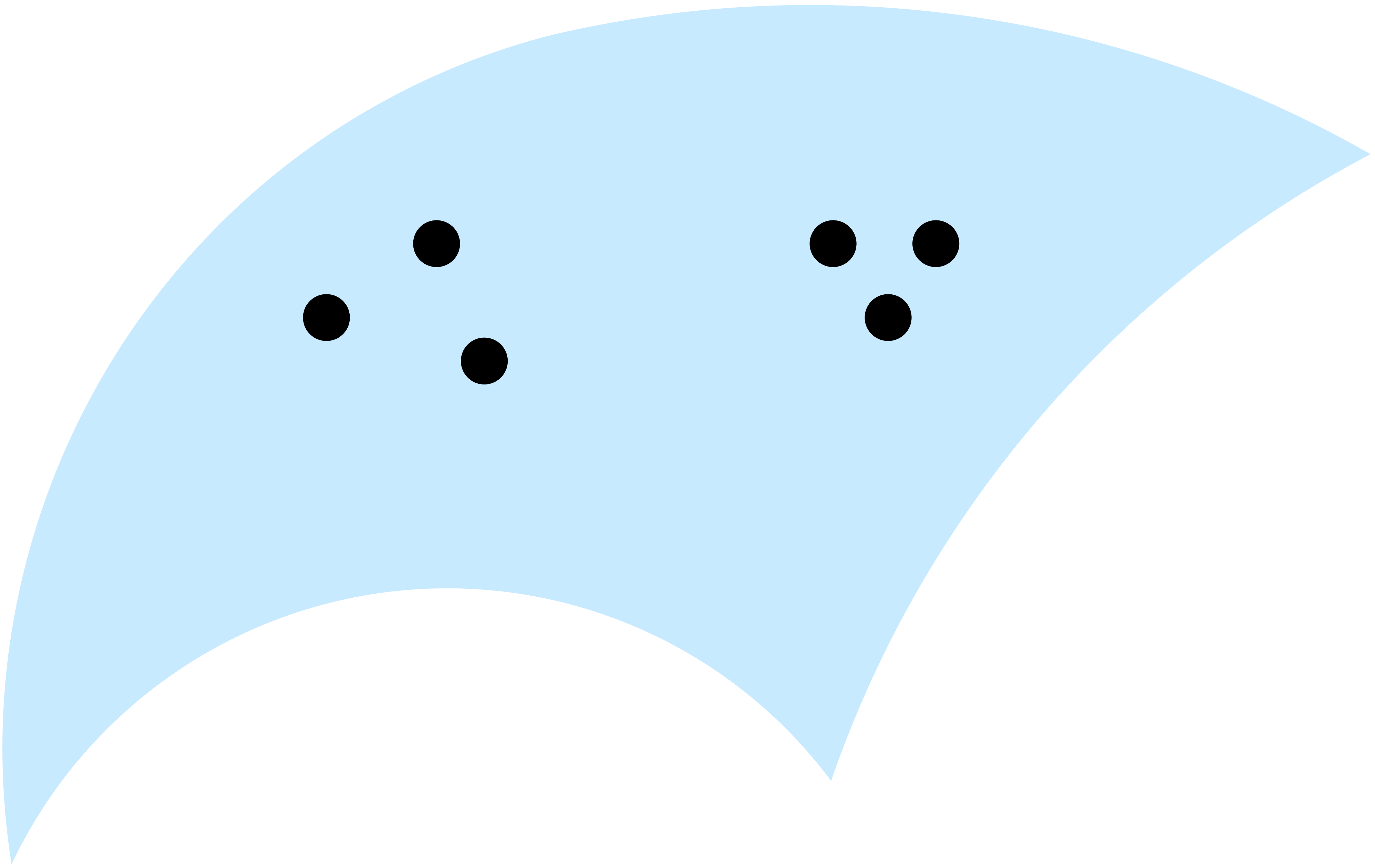
with

$$\pi^*_{nm}(G^{(t)}) = \begin{cases} \bar{w}_n & m = \text{argmin}_{m'} c(\Phi(\cdot\,;\bar{\theta}_n), \Phi(\cdot\,;\theta^{(t)}_{m'})) \\ 0 & \text{otherwise}\,. \end{cases}$$

- Closed-form solution: $G^{(t+1)} = \text{argmin}_G \mathscr{K}_c(G \mid G^{(t)})$.

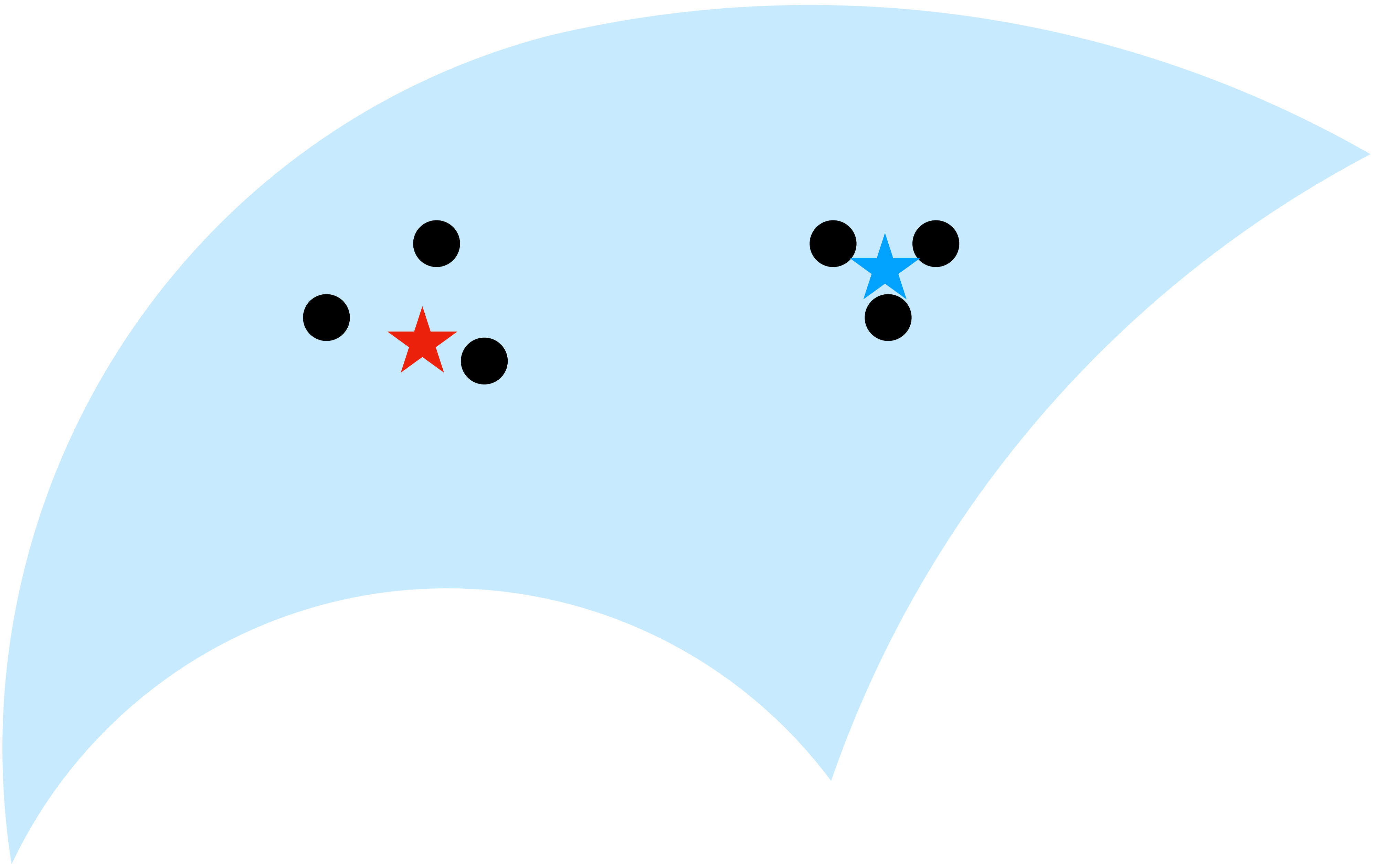
Concrete MM steps

3 machine each fit a 2 component mixture



Concrete MM steps

3 machine each fit a 2 component mixture

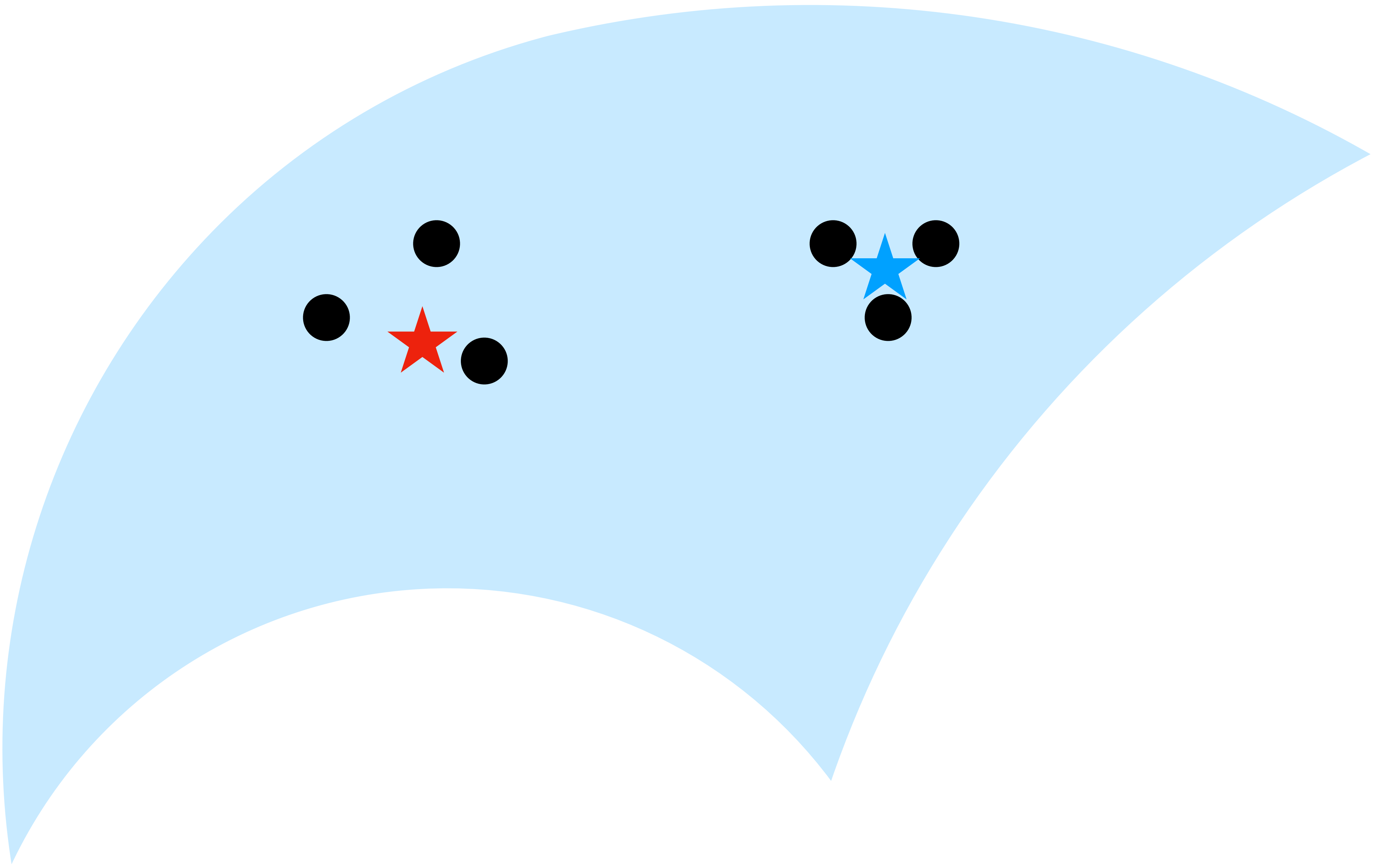


Concrete MM steps

- **Majorization step**: for a given $G^{(t)}$, the optimal transportation plan $\boldsymbol{\pi}^*(G^{(t)})$ is

$$\pi^*_{nm}(G^{(t)}) = \begin{cases} \bar{w}_n & \text{if } m = \text{argmin}_{m'} c(\bar{\Phi}_n, \Phi^{(t)}_{m'}) \\ 0 & \text{o.w.} \end{cases}$$

3 machine each fit a 2 component mixture

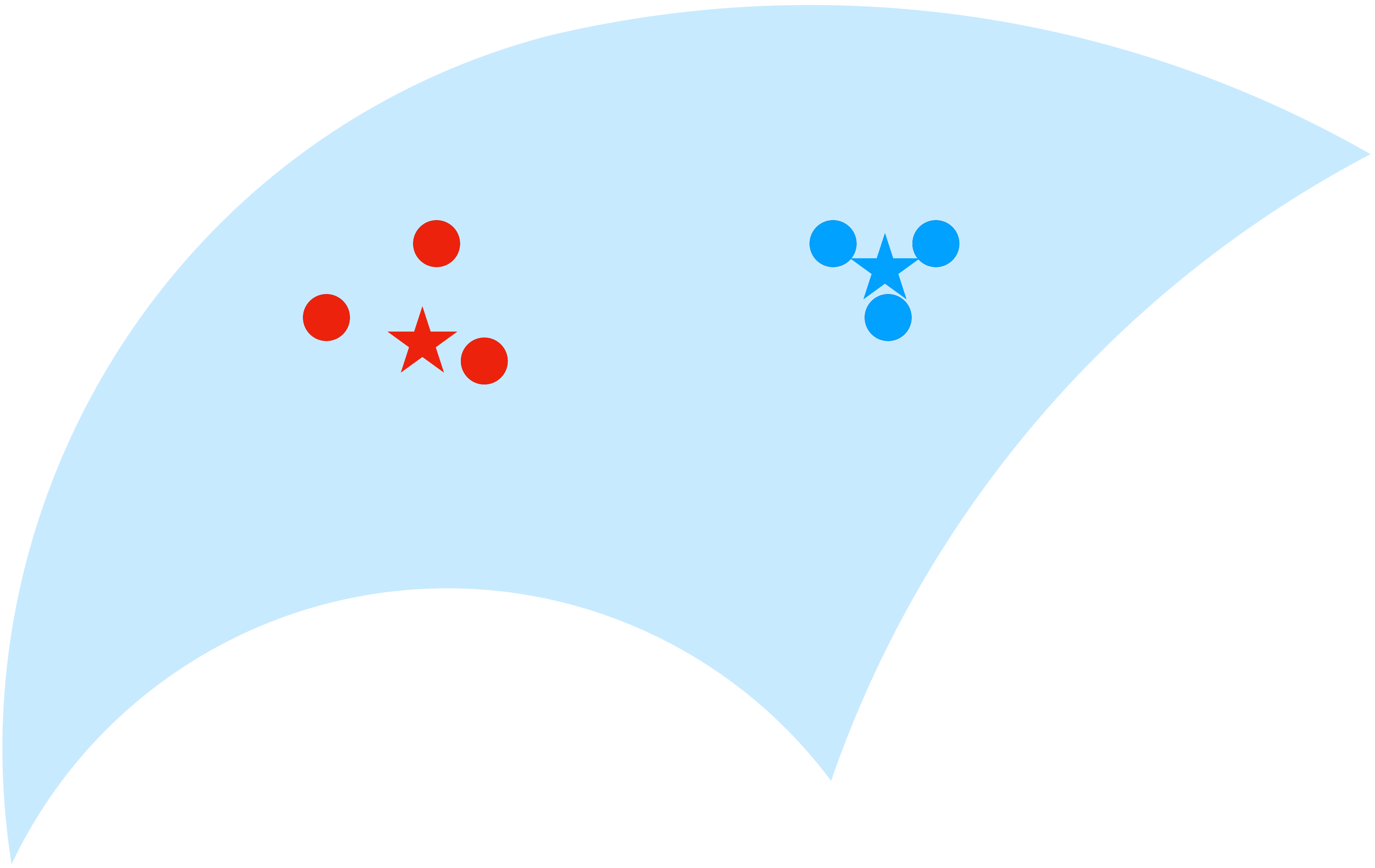


Concrete MM steps

- **Majorization step**: for a given $G^{(t)}$, the optimal transportation plan $\boldsymbol{\pi}^*(G^{(t)})$ is

$$\pi^*_{nm}(G^{(t)}) = \begin{cases} \bar{w}_n & \text{if } m = \text{argmin}_{m'} c(\bar{\Phi}_n, \Phi^{(t)}_{m'}) \\ 0 & \text{o.w.} \end{cases}$$

3 machine each fit a 2 component mixture



Concrete MM steps

- **Majorization step**: for a given $G^{(t)}$, the optimal transportation plan $\boldsymbol{\pi}^*(G^{(t)})$ is

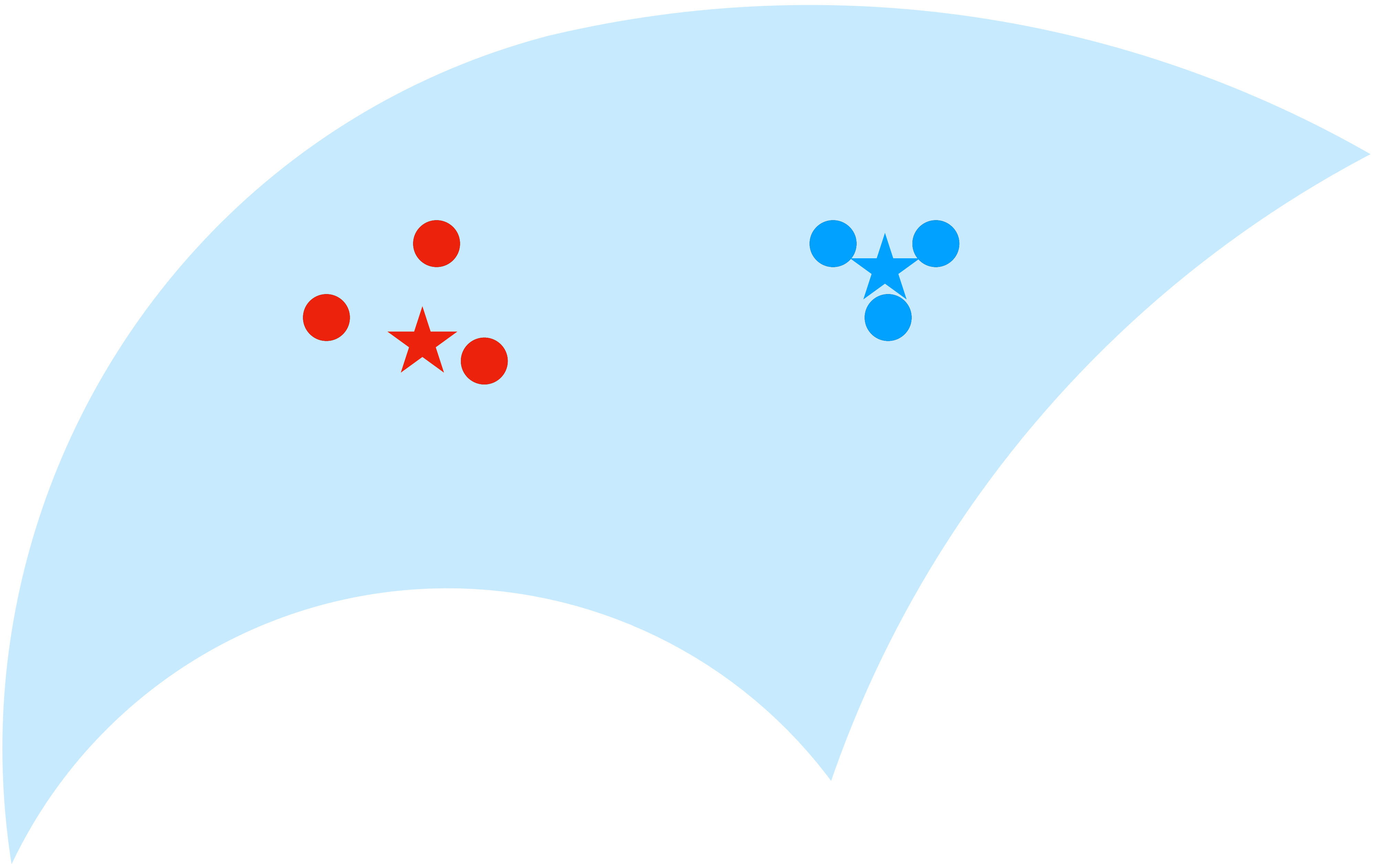
$$\pi^*_{nm}(G^{(t)}) = \begin{cases} \bar{w}_n & \text{if } m = \text{argmin}_{m'} c(\bar{\Phi}_n, \Phi^{(t)}_{m'}) \\ 0 & \text{o.w.} \end{cases}$$

- **Minimization step**: for a given $\boldsymbol{\pi}$, the subpopulation parameters are

$$\Phi^{(t+1)}_m = \text{arginf}_{\Phi} \sum_n \pi^*_{nm}(G^{(t)}) c(\bar{\Phi}_n, \Phi)$$

$$w^{(t+1)}_m = \sum_n \pi^*_{nm}(G^{(t)})$$

3 machine each fit a 2 component mixture



Concrete MM steps

- **Majorization step**: for a given $G^{(t)}$, the optimal transportation plan $\boldsymbol{\pi}^*(G^{(t)})$ is

$$\pi^*_{nm}(G^{(t)}) = \begin{cases} \bar{w}_n & \text{if } m = \text{argmin}_{m'} c(\bar{\Phi}_n, \Phi^{(t)}_{m'}) \\ 0 & \text{o.w.} \end{cases}$$

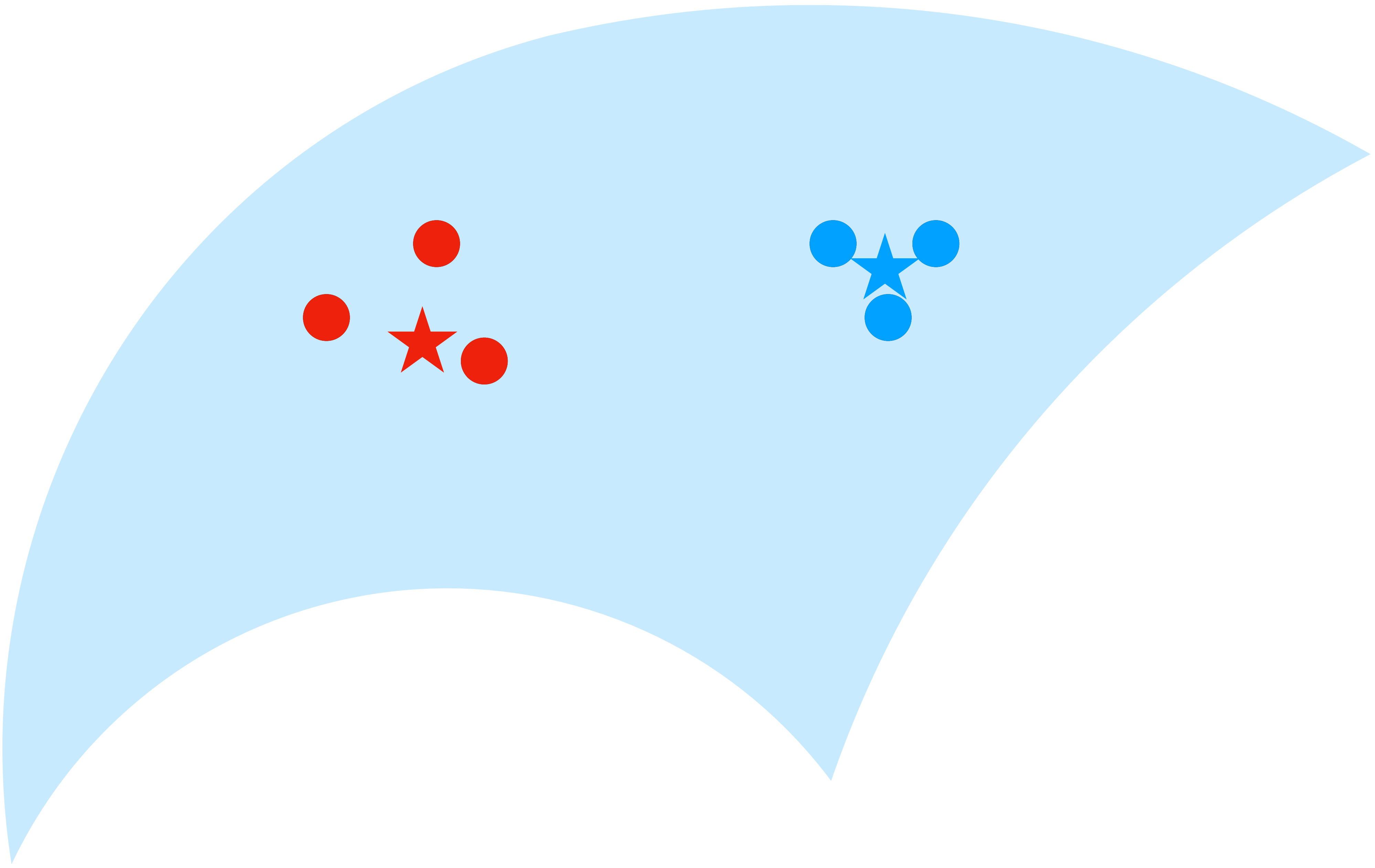
- **Minimization step**: for a given $\boldsymbol{\pi}$, the subpopulation parameters are

Barycenter of Gaussians (analytical form)

$$\boxed{\Phi^{(t+1)}_m = \text{arginf}_{\Phi} \sum_n \pi^*_{nm}(G^{(t)}) c(\bar{\Phi}_n, \Phi)}$$

$$w^{(t+1)}_m = \sum_n \pi^*_{nm}(G^{(t)})$$

3 machine each fit a 2 component mixture



Concrete MM steps

- **Majorization step**: for a given $G^{(t)}$, the optimal transportation plan $\boldsymbol{\pi}^*(G^{(t)})$ is

$$\pi^*_{nm}(G^{(t)}) = \begin{cases} \bar{w}_n & \text{if } m = \text{argmin}_{m'} c(\bar{\Phi}_n, \Phi^{(t)}_{m'}) \\ 0 & \text{o.w.} \end{cases}$$

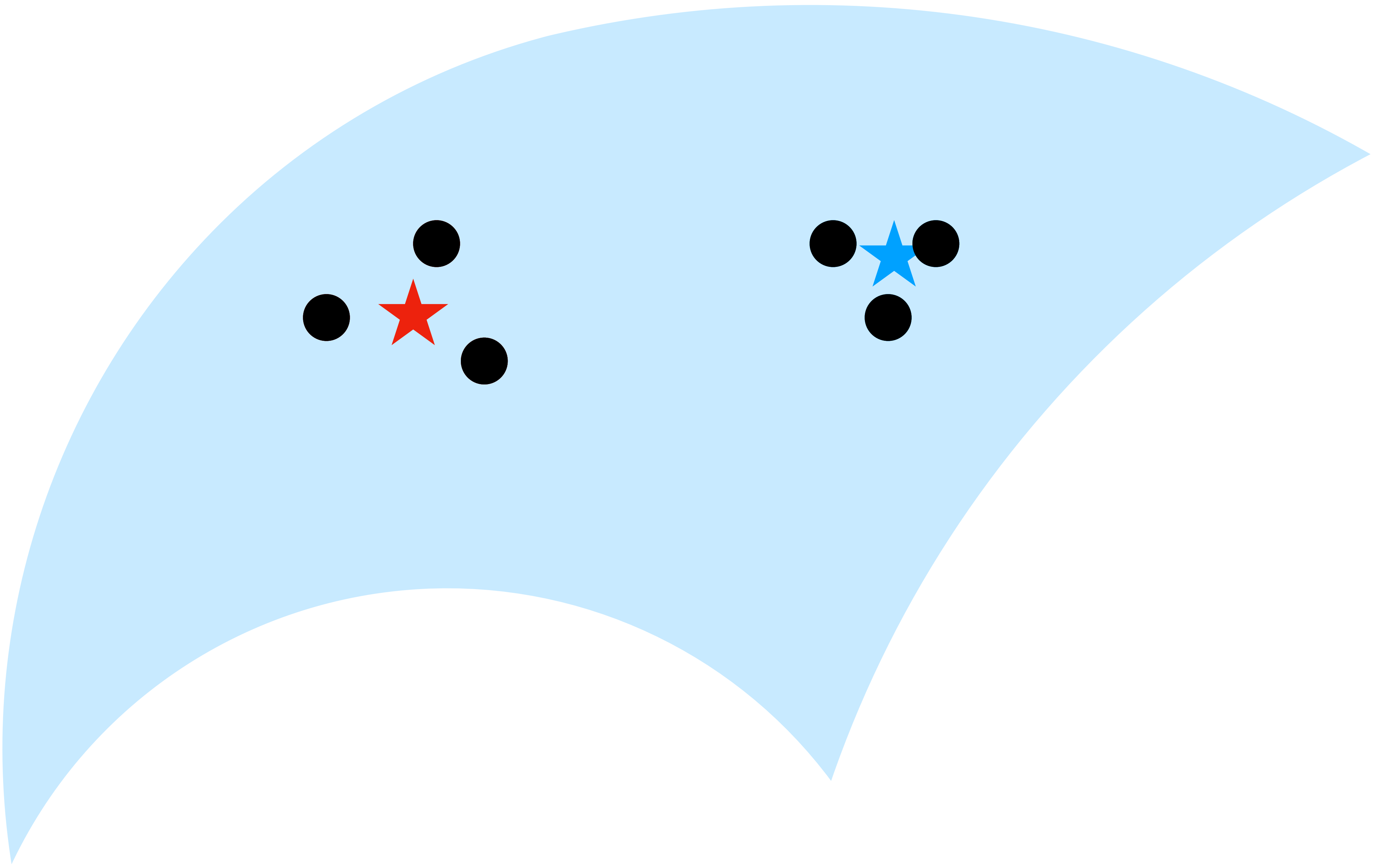
- **Minimization step**: for a given $\boldsymbol{\pi}$, the subpopulation parameters are

Barycenter of Gaussians (analytical form)

$$\boxed{\Phi^{(t+1)}_m = \text{arginf}_\Phi \sum_n \pi^*_{nm}(G^{(t)}) c(\bar{\Phi}_n, \Phi)}$$

$$w^{(t+1)}_m = \sum_n \pi^*_{nm}(G^{(t)})$$

3 machine each fit a 2 component mixture



Concrete MM steps

- **Majorization step**: for a given $G^{(t)}$, the optimal transportation plan $\boldsymbol{\pi}^*(G^{(t)})$ is

$$\pi^*_{nm}(G^{(t)}) = \begin{cases} \bar{w}_n & \text{if } m = \text{argmin}_{m'} c(\bar{\Phi}_n, \Phi^{(t)}_{m'}) \\ 0 & \text{o.w.} \end{cases}$$

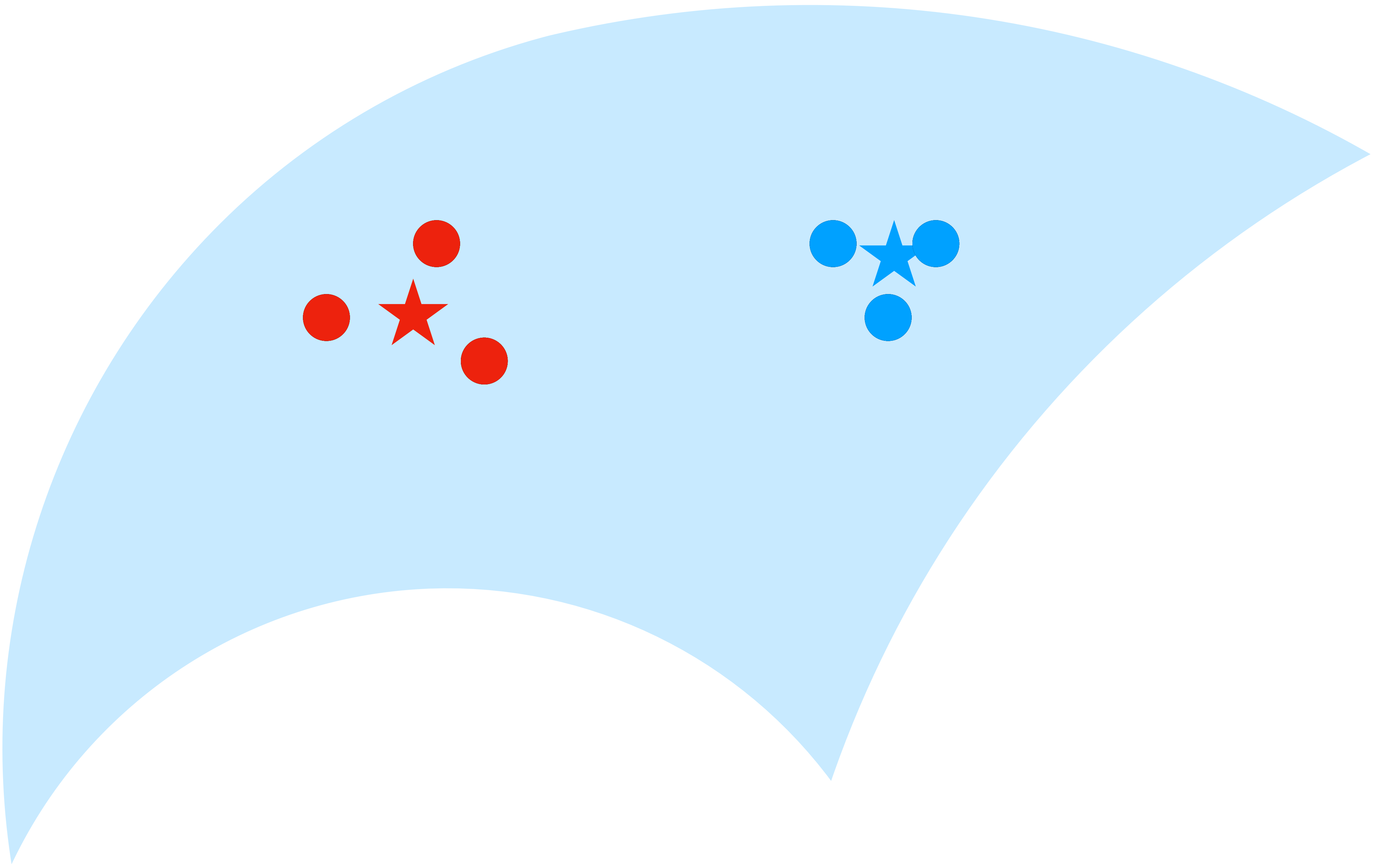
- **Minimization step**: for a given $\boldsymbol{\pi}$, the subpopulation parameters are

Barycenter of Gaussians (analytical form)

$$\boxed{\Phi^{(t+1)}_m = \text{arginf}_{\Phi} \sum_n \pi^*_{nm}(G^{(t)}) c(\bar{\Phi}_n, \Phi)}$$

$$w^{(t+1)}_m = \sum_n \pi^*_{nm}(G^{(t)})$$

3 machine each fit a 2 component mixture



Algorithm convergence

Suppose the **cost function** $c(\cdot,\cdot)$ is continuous in both arguments. For any constant $\Delta > 0$ and Φ^* the following set is compact:

$$\{\Phi : c(\Phi, \Phi^*) \leq \Delta\} .$$

Then

(i)$\mathcal{J}_c(G^{(t+1)}) \leq \mathcal{J}_c(G^{(t)})$ for any t.

(ii) if G^* is a limiting point of $G^{(t)}$, then $G^{(t)} = G^*$ implies $\mathcal{J}_c(G^{(t+1)}) = \mathcal{J}_c(G^*)$.

Our full recipe

1. Obtain local estimates $\hat{G}_m$
2. Form plain average $\bar{G} = \sum_m \lambda_m \hat{G}_m$
3. Choose CTD

$$\rho(\bar{G}, G) = \min \left\{ \sum_{n,m} \pi_{nm} \mathsf{D}_{\mathsf{KL}}(\Phi(\,\cdot\,; \bar{\theta}_n) \| \Phi(\,\cdot\,; \theta_m)) : \sum_n \pi_{nm} = w_m, \sum_m \pi_{nm} = \bar{w}_n \right\}$$

4. Use MM algorithm to find $\bar{G}^R$

Statistical assurance

C1 The data are IID observations from $\Phi(x; G^*)$ with order K

C3 The local machine sample ratios $\lambda_m = N_m/N$ have nonzero limits as $N \to \infty$

C5 The **cost** function satisfies local triangular inequality

$$A^{-1}\|\Phi_1 - \Phi_2\|^2 \leq c(\Phi_1, \Phi_2) \leq A\|\Phi_1 - \Phi_2\|^2$$

Under conditions C1-C5, up to permutations, we have

$$\bar{\Phi}^R - \Phi_k^* = O_p(N^{-1/2}), \quad \bar{w}^R - w_k^* = O_p(N^{-1/2})$$

Numerical results

Simulation setting

- Generate 100 random Gaussian mixtures of dimension $d = 50$ and $K = 5$
- We set the "degree of overlap" (MaxOmega) between subpopulation to be 1%, 5%, 10%

$$\text{MaxOmega} = \max_{i,j \in [K]} \{o_{j|i} + o_{i|j}\}$$

where $o_{j|i} = \mathbb{P}(w_i \phi(X; \theta_i) < w_j \phi(X; \theta_j) \,|\, X \sim f(\,\cdot\,; \theta_i))$ is the pairwise overlap

- Total sample size $N = 2^{21}$ ($\sim 10^6$)
- The number of local machines are set to $M = 4,\ 16,\ 64$

Estimators for comparison

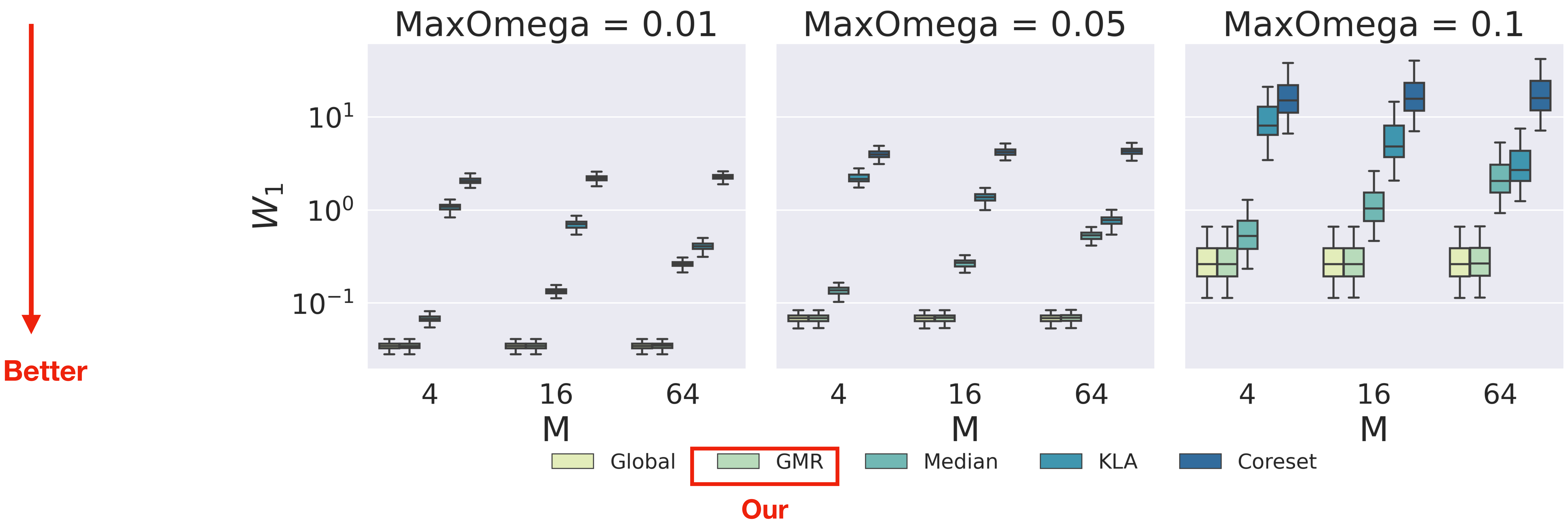
- ***Global***: the estimator based on the full dataset
- ***Median***: the “best” local estimator
- ***Reduction***: our method with KL divergence as cost function
- ***KLA*** (Liu et al. 2013)
- ***Coreset*** (Lucic et al. 2018)

Estimators for comparison

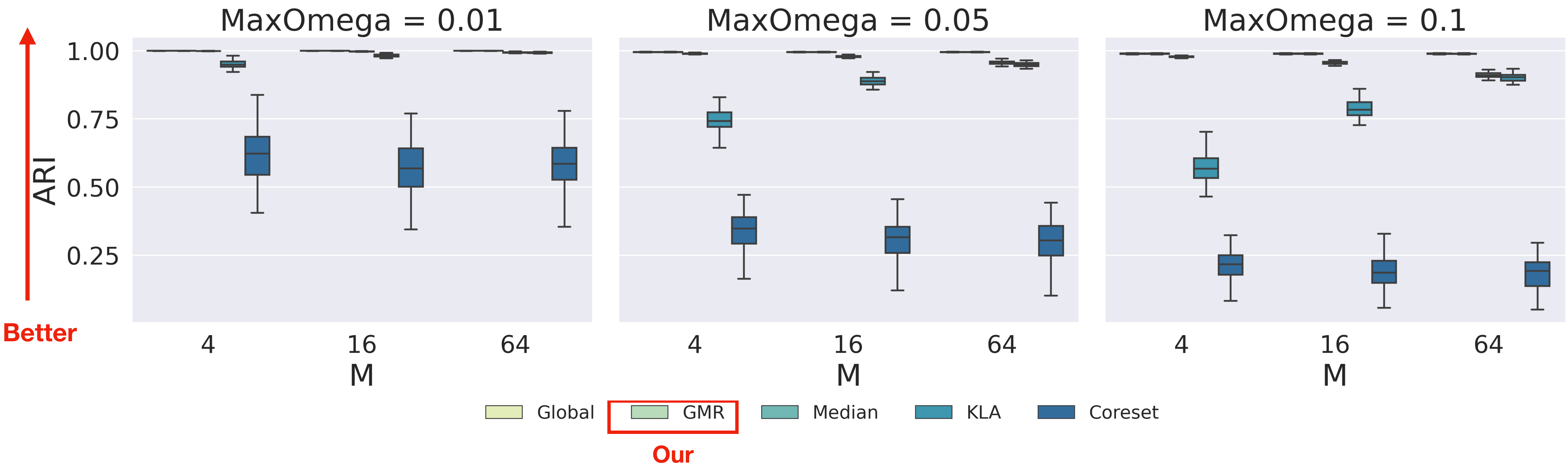
- ***Global***: the estimator based on the full dataset
- ***Median***: the “best” local estimator
- ***Reduction***: our method with KL divergence as cost function
- ***KLA*** (Liu et al. 2013)
- ***Coreset*** (Lucic et al. 2018)

Existing methods in literature

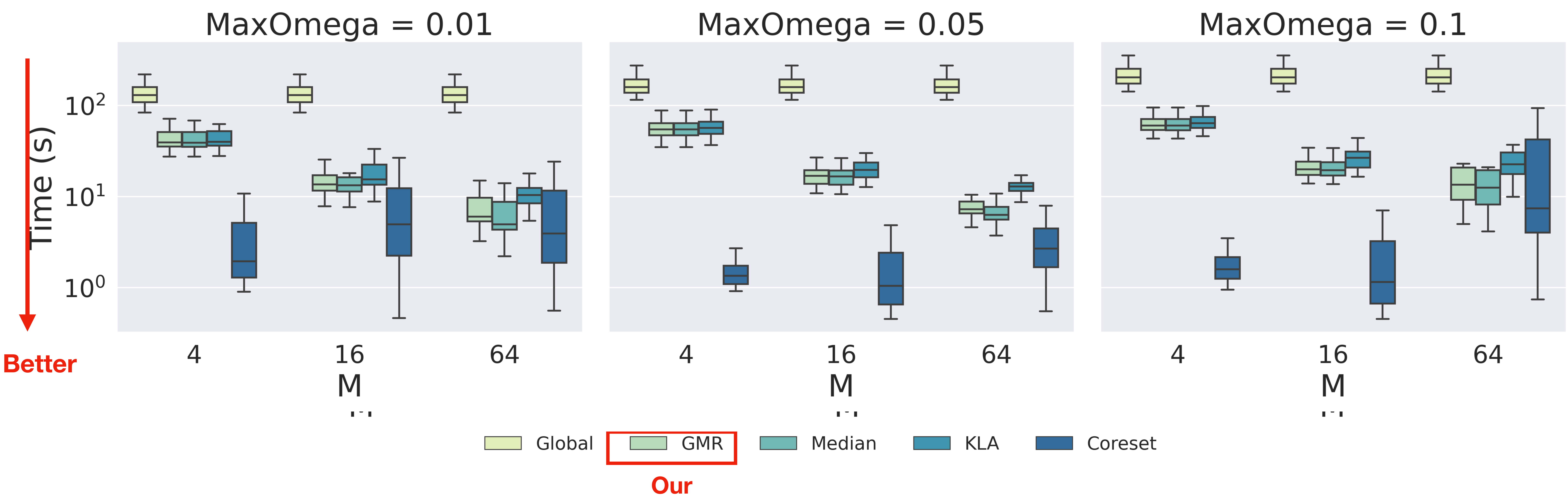
Simulation results



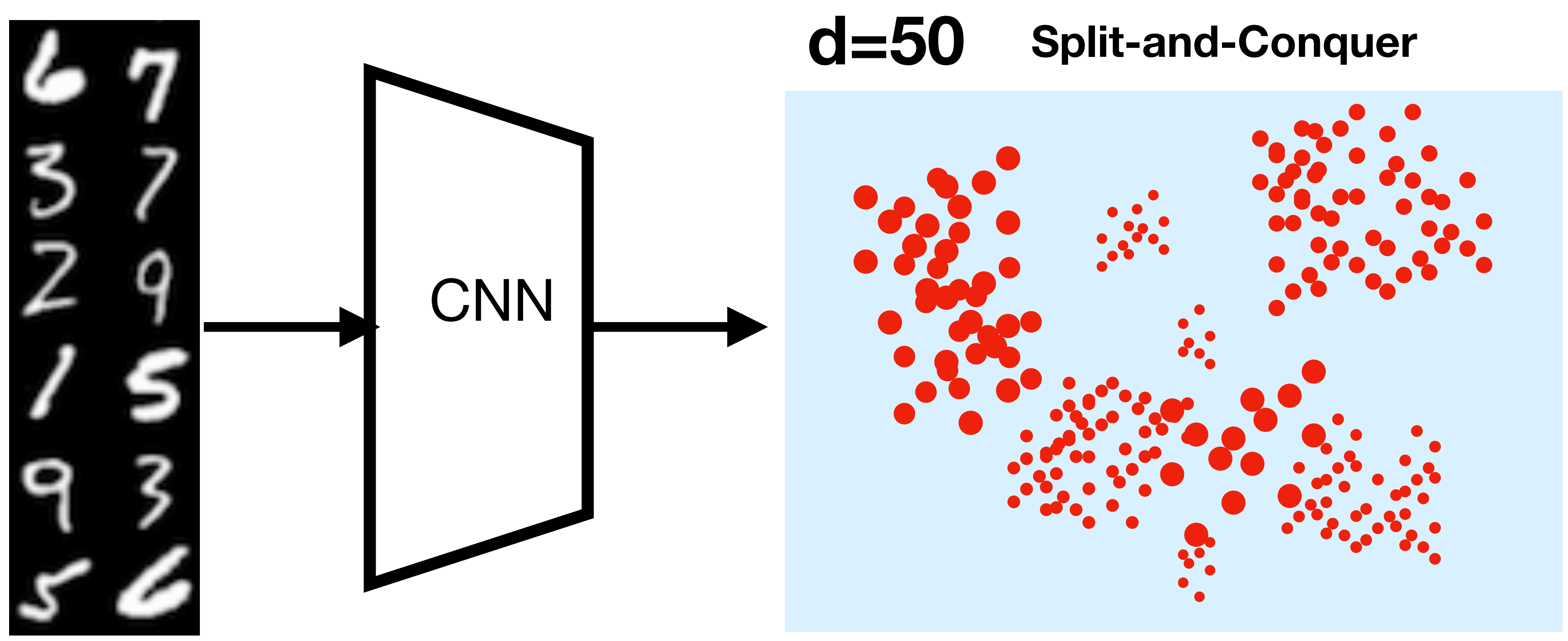
Simulation results



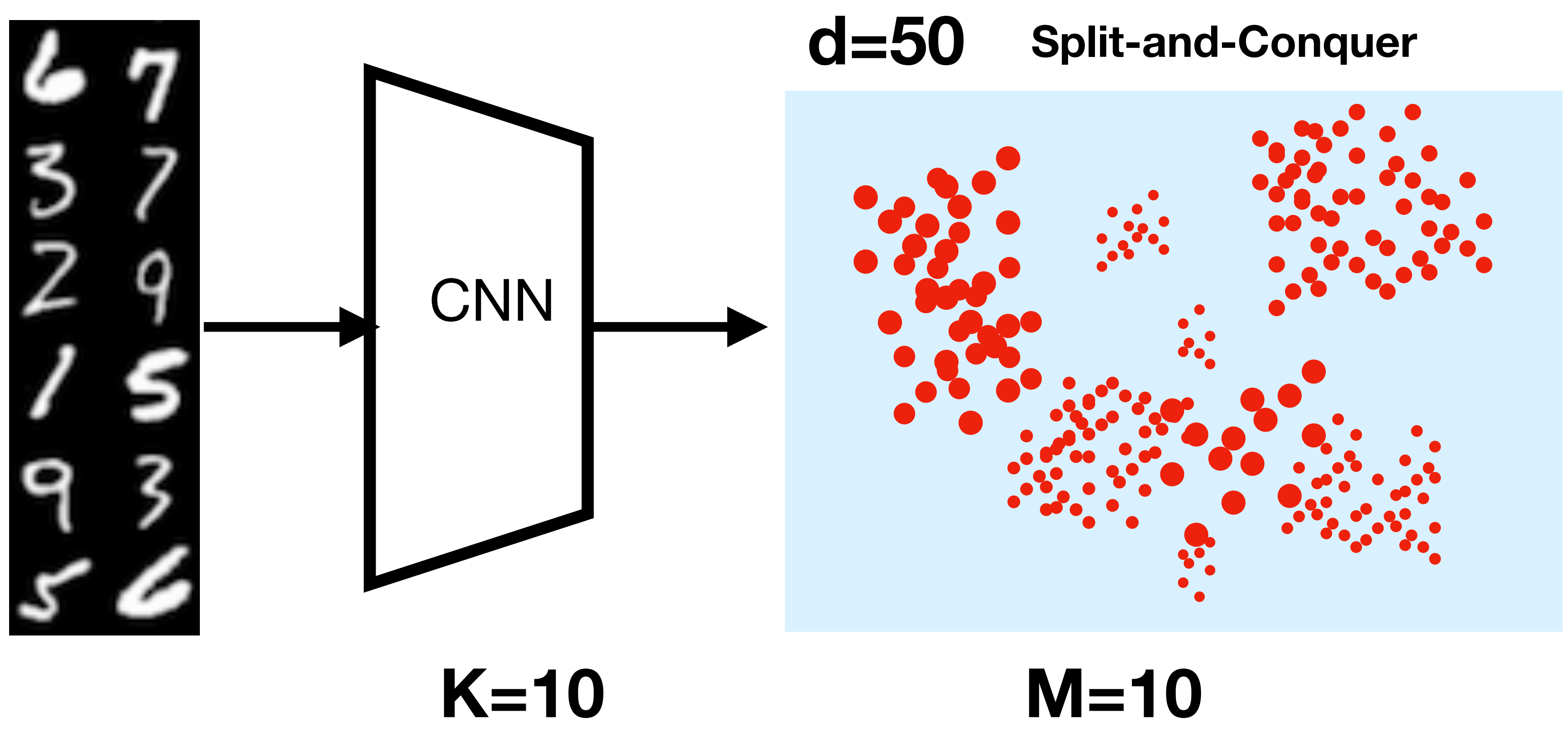
Simulation results



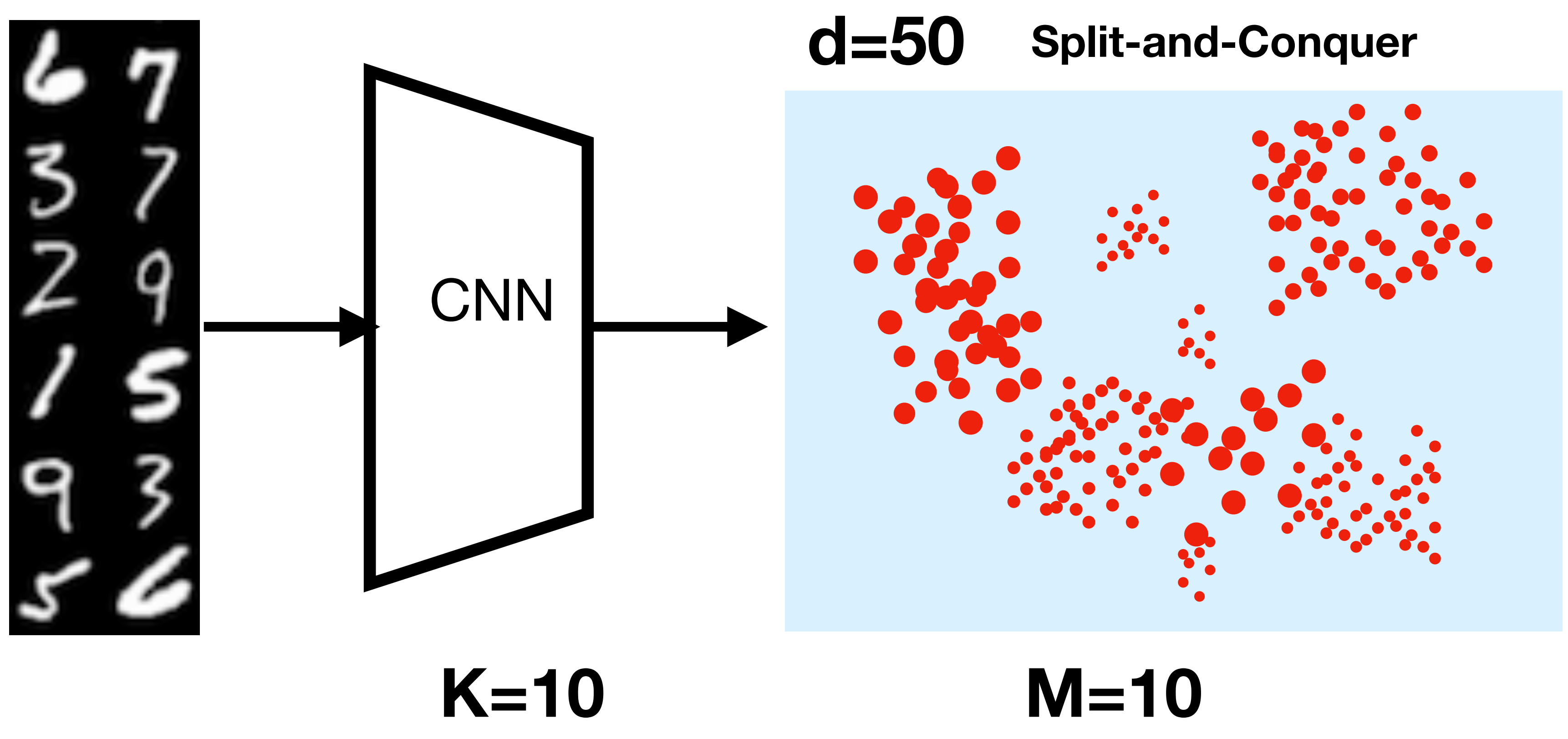
Real data: NIST clustering



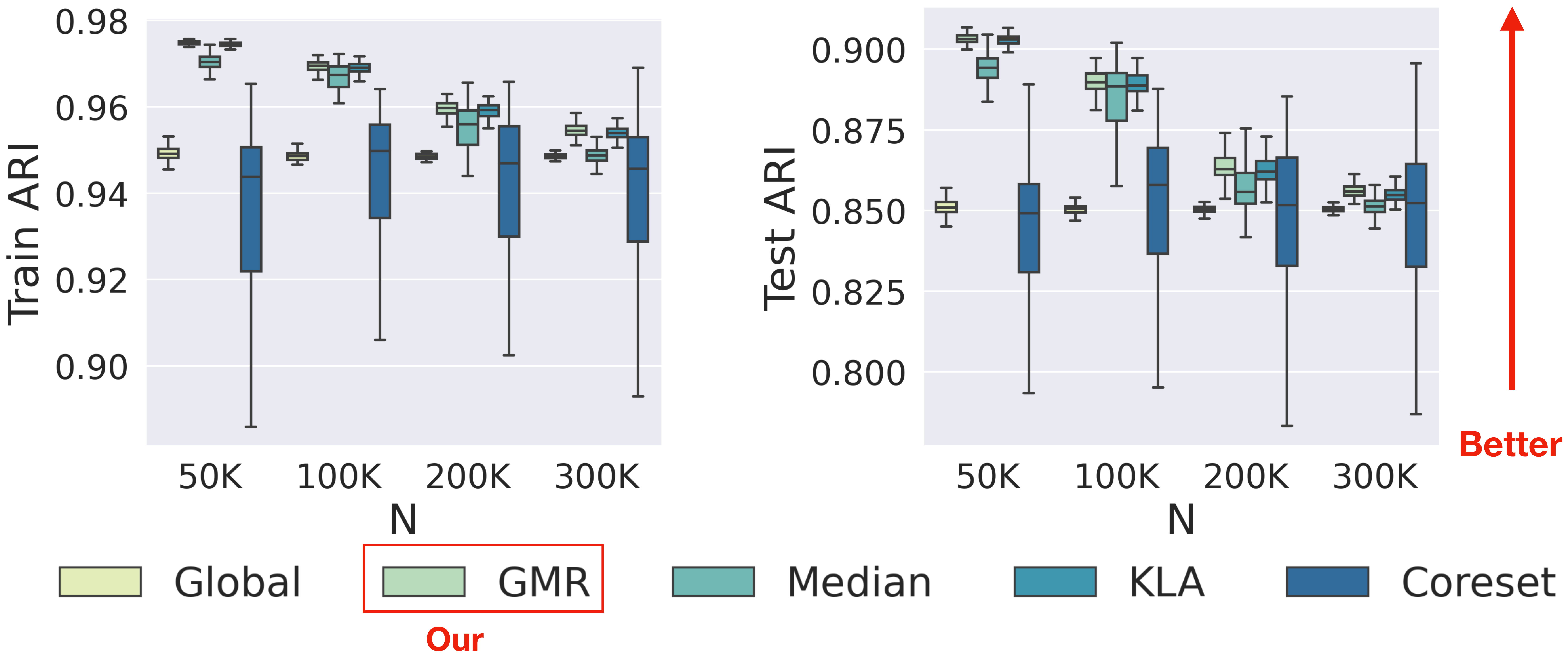
Real data: NIST clustering



Real data: NIST clustering



ARI: similarity between true label vs predicted cluster based on fitted mixture



Link to our paper

Summary of our contribution

- Developed a novel aggregation method for split-and-conquer learning of finite mixture models.
- Theoretically shown the aggregated estimator is
  - computationally efficient.
  - root-n consistent when the order is known.
- Empirically demonstrated the superior performance of the proposed estimator.

Link to our paper

Summary of our contribution

- Developed a novel aggregation method for split-and-conquer learning of finite mixture models.
- Theoretically shown the aggregated estimator is
  - computationally efficient.
  - root-n consistent when the order is known.
- Empirically demonstrated the superior performance of the proposed estimator.

Thank you!