Robustly Learning of Arbitrary Gaussian Mixtures


Ainesh Bakshi


Ilias Diakonikolas

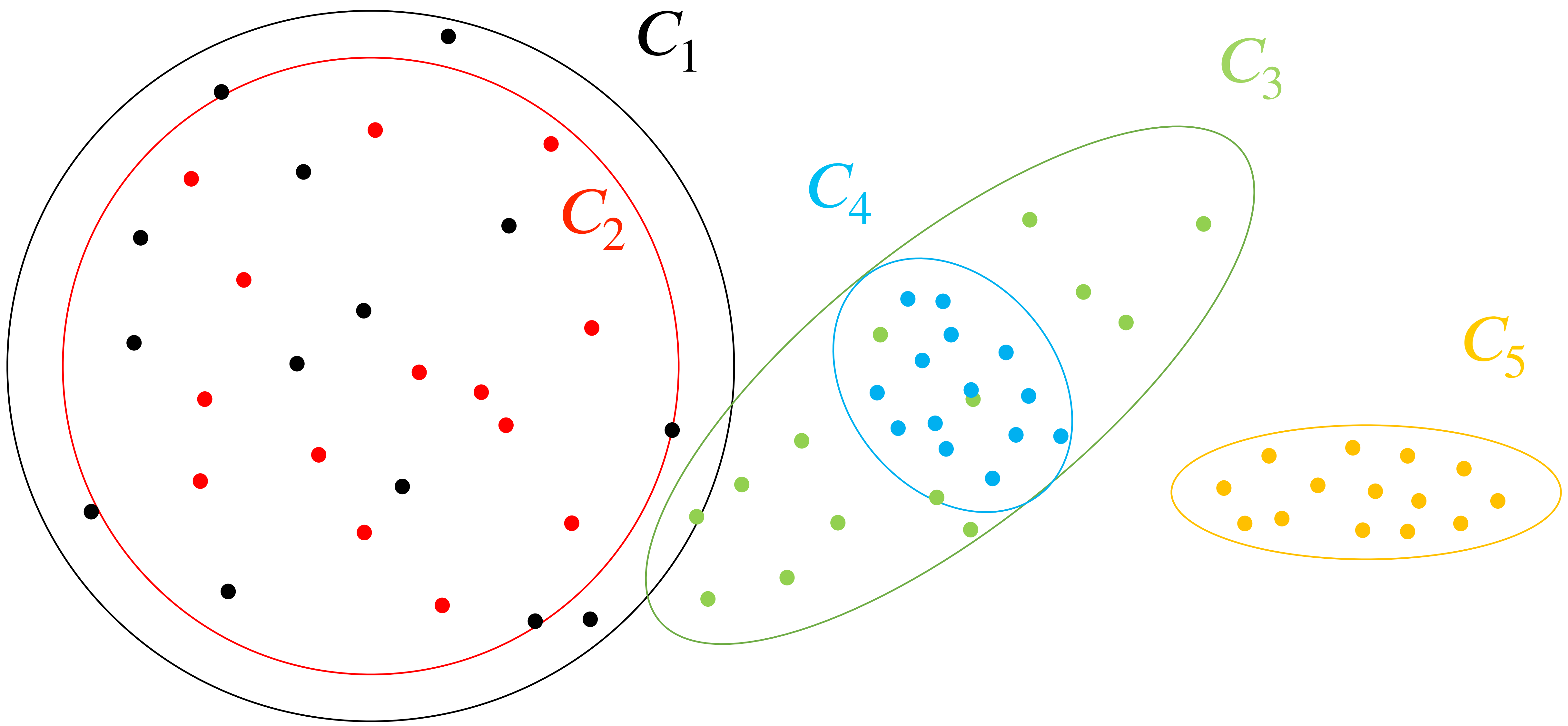
He Jia


Daniel Kane


Pravesh K. Kothari


Santosh Vempala

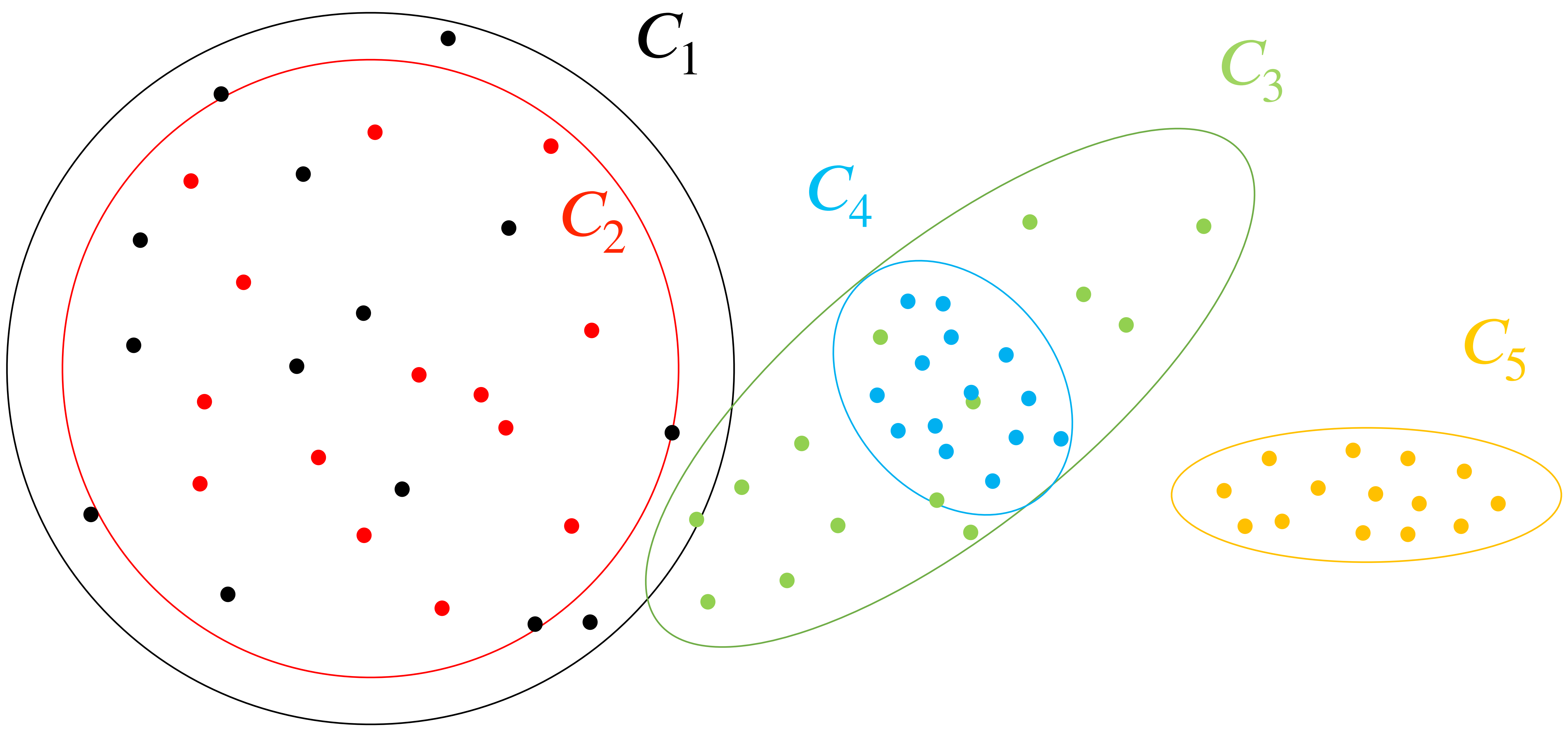
Gaussian Mixture Models



- Mixtures of $k = 5$ Gaussians in $\mathbb{R}^d$: with probability w_i, sample from $N(\mu_i, \Sigma_i)$
- d: dimension
- k: number of components
- w_i: weights
- μ_i: means
- Σ_i: covariances

Learning Gaussian Mixture Models

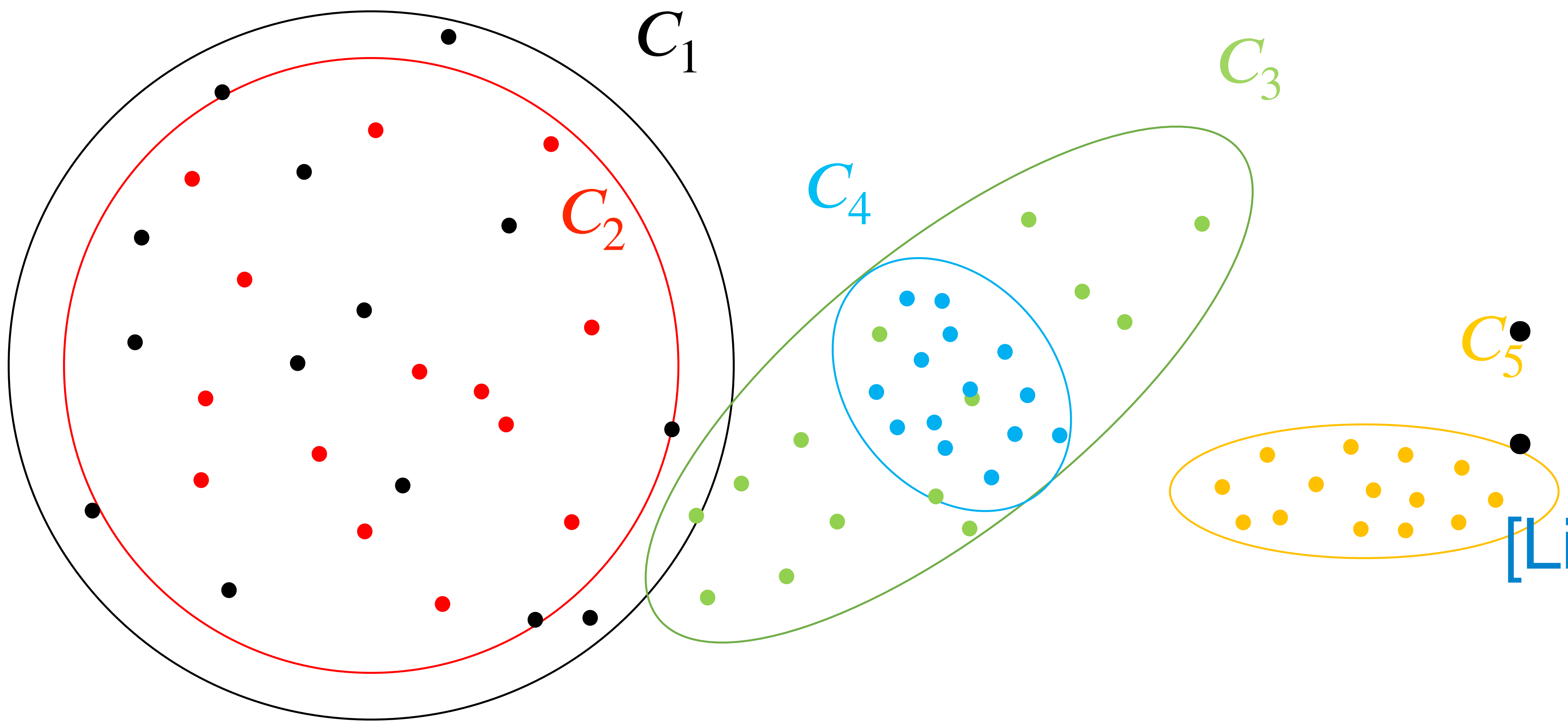
- Input: i.i.d. samples from a Gaussian mixture M
- Output: A Gaussian mixture $\widehat{M}$ close to M in total variation distance



Learning Gaussian Mixture Models

- Input: i.i.d. samples from a Gaussian mixture M
- Output: A Gaussian mixture $\widehat{M}$ close to M in total variation distance

$$d_{TV}(p,q) = \frac{1}{2}\int |p(x) - q(x)|\, dx$$



- natural info-theoretic measure
- implies all parameter distance guarantees

[Liu-Moitra'21,Bakshi-Diakonikolas-J-Kane-Kothari-Vempala'22]

Learning Gaussian Mixture Models

- Input: i.i.d. samples from a Gaussian mixture M
- Output: A Gaussian mixture $\widehat{M}$ close to M in total variation distance

[Pearson 1894]

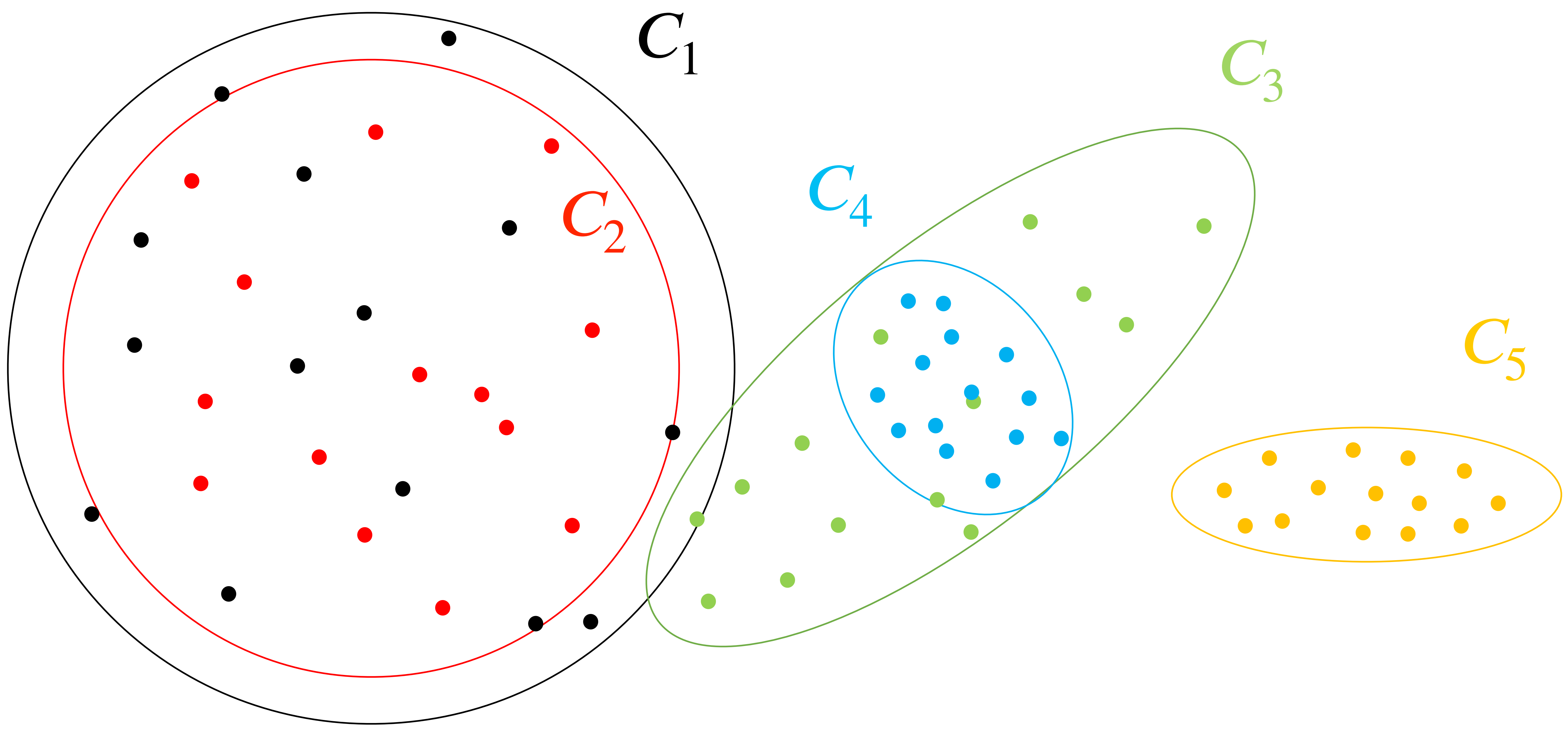
...

[Dasgupta'98]	**Random Projection**
[Arora-Kannan'01]	
[Vempala-Wang'02]	**PCA**
[Brubaker-Vempala'08]	**Isotropic PCA**
...	

[Kalai-Moitra-Valiant'09, Moitra-Valiant'10] **Many 1-d Random Projections**
[Belkin-Sinha'10]

Learning Gaussian Mixture Models

- [Moitra-Valiant'10, *based on* Kalai-M-V'09]: There is an algorithm that learns k-GMMs up to δ-TV error in time $(d/\delta)^{k^{O(k^2)}}$

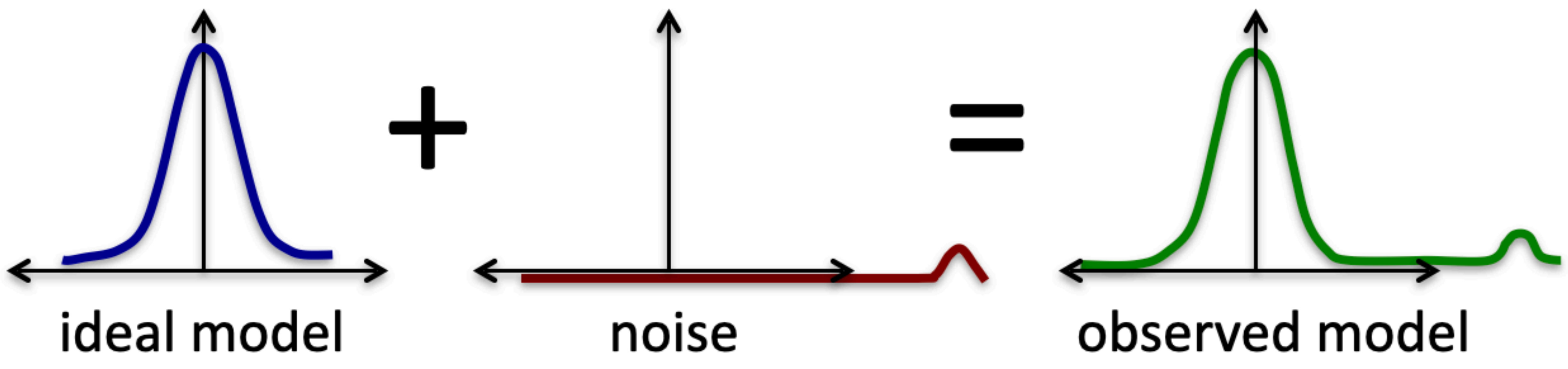


What if Data has Outliers?

- Robust statistical models
  - Tukey and Huber initiated work in 60's
  - Capture systematic error and adversarial outliers

Robust Learning

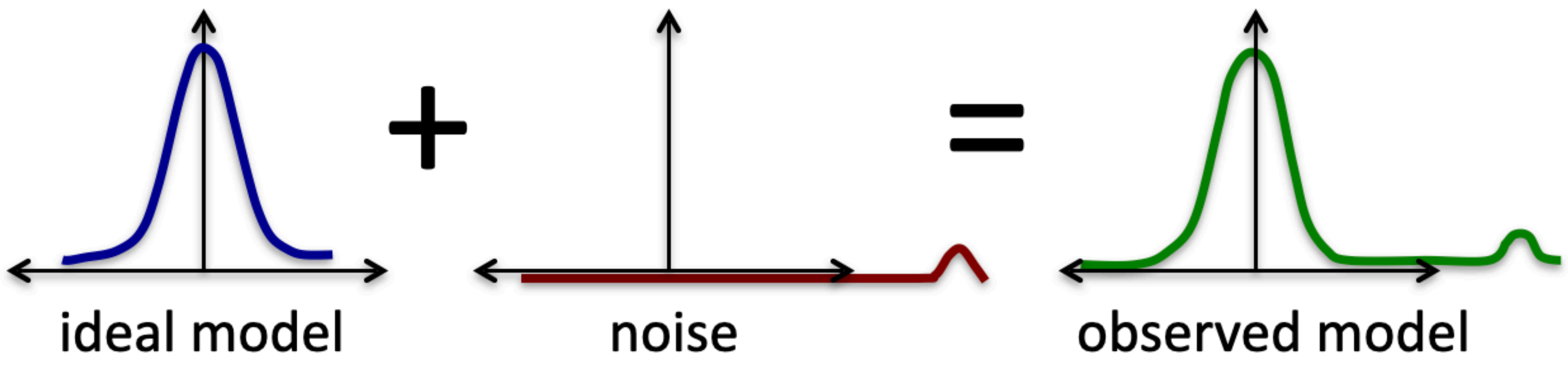
- Input: a constant fraction ϵ of data is arbitrarily corrupted by the adversary
  - We know nothing about the corrupted data, except the number is bounded
- Goal: learn the distribution within total variation distance
  - The error should be independent of dimension
  - The optimal error is $O(\epsilon)$



Robust Learning

- Input: a constant fraction ϵ of data is arbitrarily corrupted by the adversary
  - We know nothing about the corrupted data, except the number is bounded
- Goal: learn the distribution within total variation distance
  - The error should be independent of dimension
  - The optimal error is $O(\epsilon)$

captures the power of adversarial corruption



Robustly Learning Gaussian Mixture Models

- Input: ϵ-corrupted samples from a Gaussian mixture M
- Output: A Gaussian mixture $\widehat{M}$ $\mathrm{poly}(\epsilon)$-close to M in Total Variation Distance

- [Moitra-Valiant'10] can only handle $1/\mathrm{poly}(d)$ fraction of outliers

Robustly Learning Gaussian Mixture Models

- Theorem [Bakshi-Diakonikolas-J-Kane-Kothari-Vempala'22]: An ϵ-robust algorithm for learning arbitrary GMMs
  - Samples: $n \geq d^{O(k)}\text{poly}_k(1/\epsilon)$
  - Time: $\text{poly}(n)$
  - Error: $\text{poly}_k(\epsilon)$ in TV distance

Robustly Learning Gaussian Mixture Models

- Theorem [Bakshi-Diakonikolas-J-Kane-Kothari-Vempala'22]: An ϵ-robust algorithm for learning arbitrary GMMs
  - Samples: $n \geq d^{O(k)} \mathrm{poly}_k(1/\epsilon)$
  - Time: $\mathrm{poly}(n)$
  - Error: $\mathrm{poly}_k(\epsilon)$ in TV distance

No constraint on minimum weight/ covariances

Robustly Learning Gaussian Mixture Models

- Theorem [Bakshi-Diakonikolas-J-Kane-Kothari-Vempala'22]: An ϵ-robust algorithm for learning arbitrary GMMs
  - Samples: $n \geq d^{O(k)} \mathrm{poly}_k(1/\epsilon)$
  - Time: $\mathrm{poly}(n)$
  - Error: $\mathrm{poly}_k(\epsilon)$ in TV distance

Matches SQ lower bound in [Diakonikolas-Kane-Stewart'18]

Robustly Learning Gaussian Mixture Models

- Theorem [Bakshi-Diakonikolas-J-Kane-Kothari-Vempala'22]: An ϵ-robust algorithm for learning arbitrary GMMs
  - Samples: $n \geq d^{O(k)}\mathrm{poly}_k(1/\epsilon)$
  - Time: $\mathrm{poly}(n)$
  - Error: $\mathrm{poly}_k(\epsilon)$ in TV distance

Improves the non-robust running time of [Moitra-Valiant'10] if $\epsilon = \omega(1/d)$

Robustly Learning Gaussian Mixture Models

- Theorem [Bakshi-Diakonikolas-J-Kane-Kothari-Vempala’22]: An ϵ-robust algorithm for learning arbitrary GMMs

Samples: $n \geq d^{O(k)}\mathrm{poly}_k(1/\epsilon)$	Time: $\mathrm{poly}(n)$	Error: $\mathrm{poly}_k(\epsilon)$

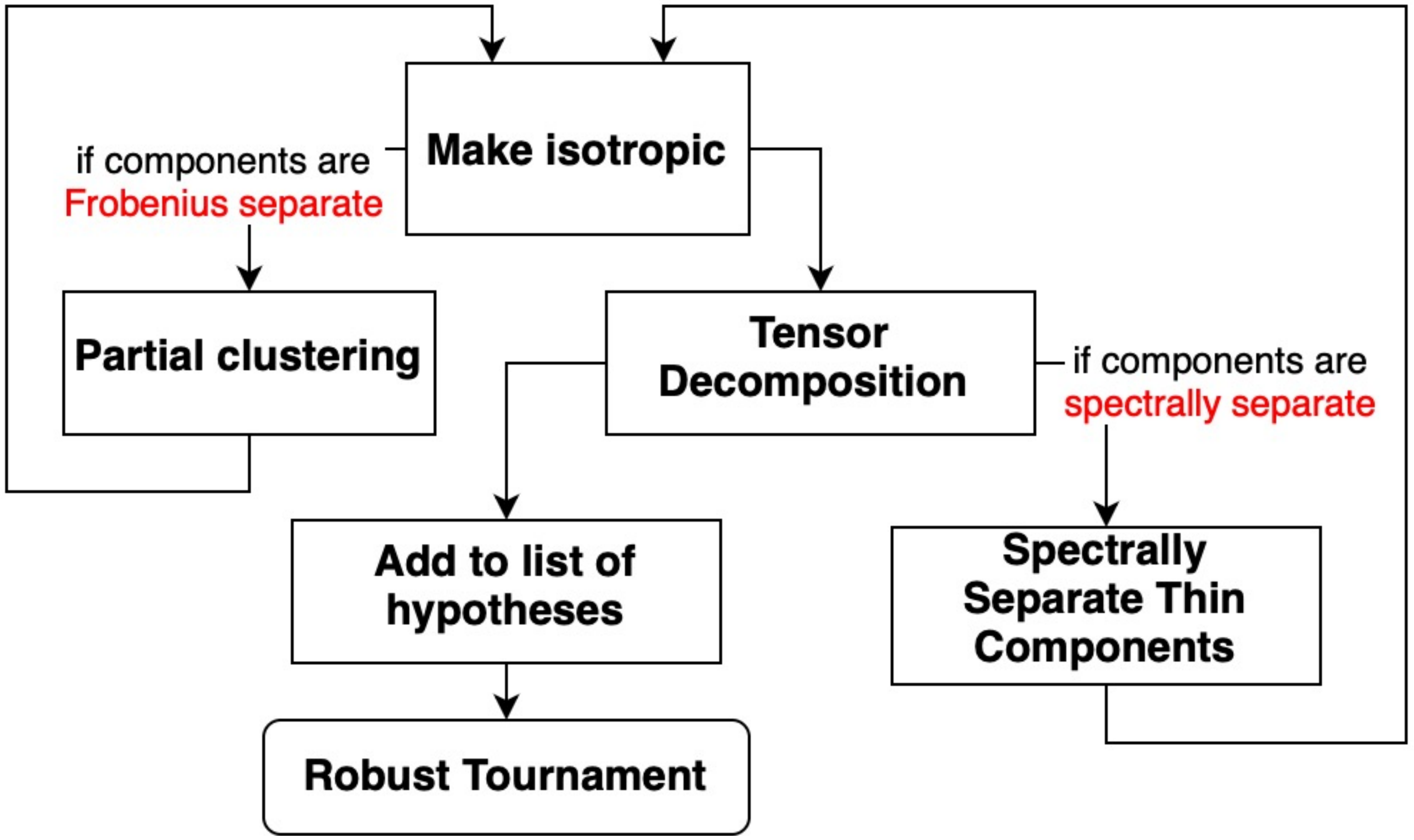
- Concurrent work [Liu-Moitra’21]: An ϵ-robust parameter estimation algorithm for GMMs s.t. all pairs $\geq \Omega_k(1)$-TV far

Samples: $n \geq d^{f(\frac{1}{w_{\min}})}\mathrm{poly}_k(1/\epsilon)$	Time: $\mathrm{poly}(n)$	Error: $\epsilon^{f(\frac{1}{w_{\min}})}$

Parameter Recovery

- Theorem: ϵ-TV distance implies $\mathrm{poly}_k(\epsilon)$-component distance for arbitrary Gaussian mixtures
  - Relies on a key lemma from [Liu-Moitra'21]
  - Generalizes the identifiability theorem in [Liu-Moitra'21]
- Corollary [Bakshi-Diakonikolas-J-Kane-Kothari-Vempala'22]: The same algorithm in our main theorem also recovers the parameters/components.

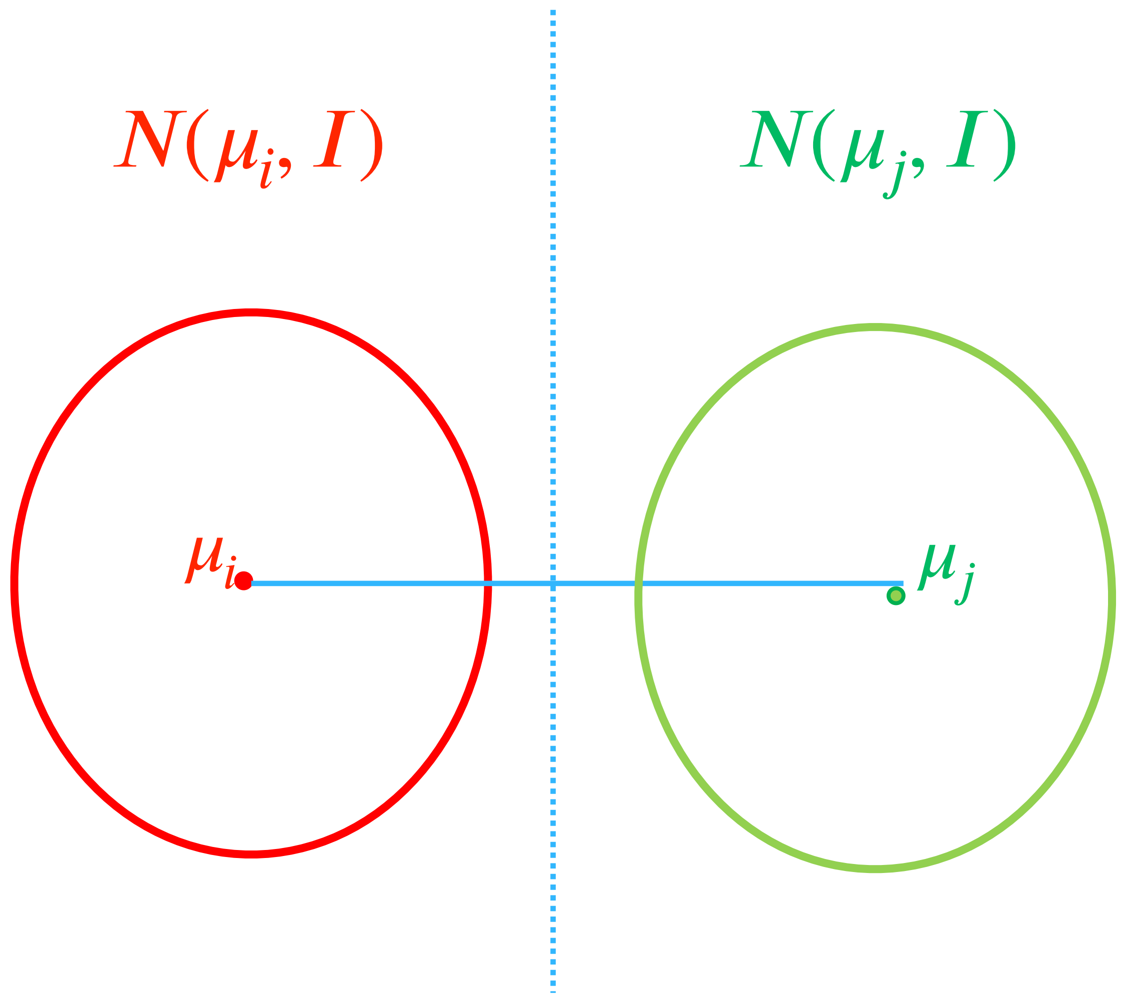
Overall Algoritm



TV Distance Separation

- Two Gaussians are separated in TV distance iff one of the following holds:

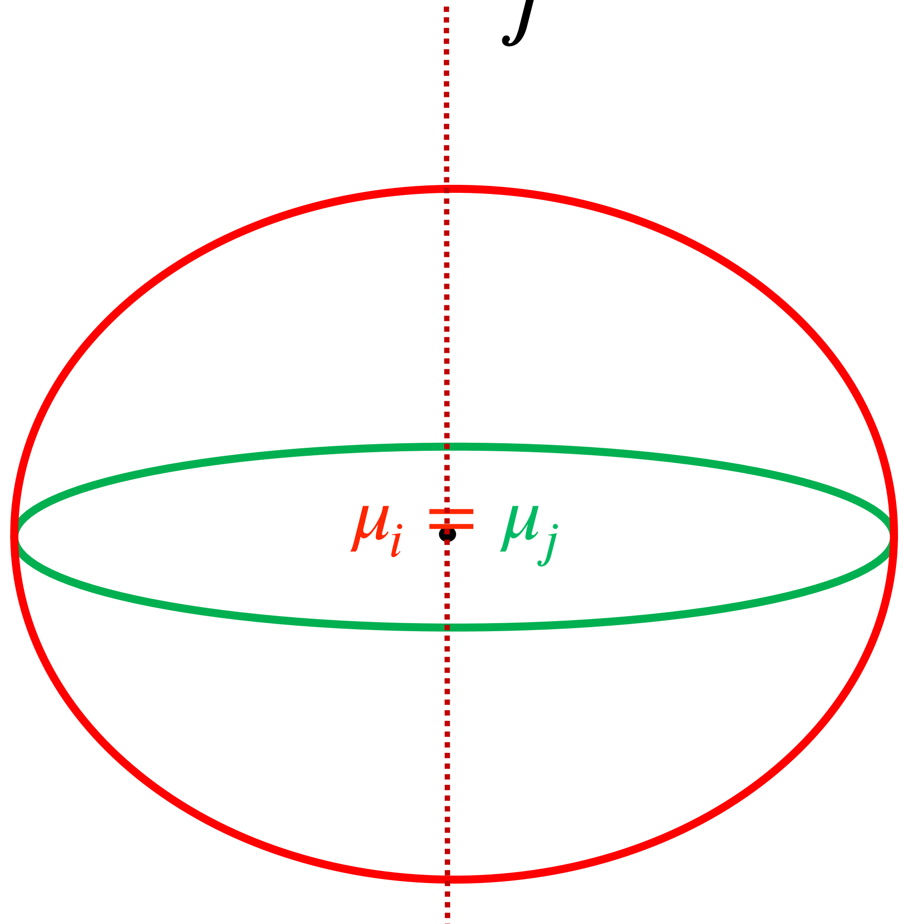
Mean-separated



e.g., $\left|\left|\mu_i - \mu_j\right|\right| \gg 2$

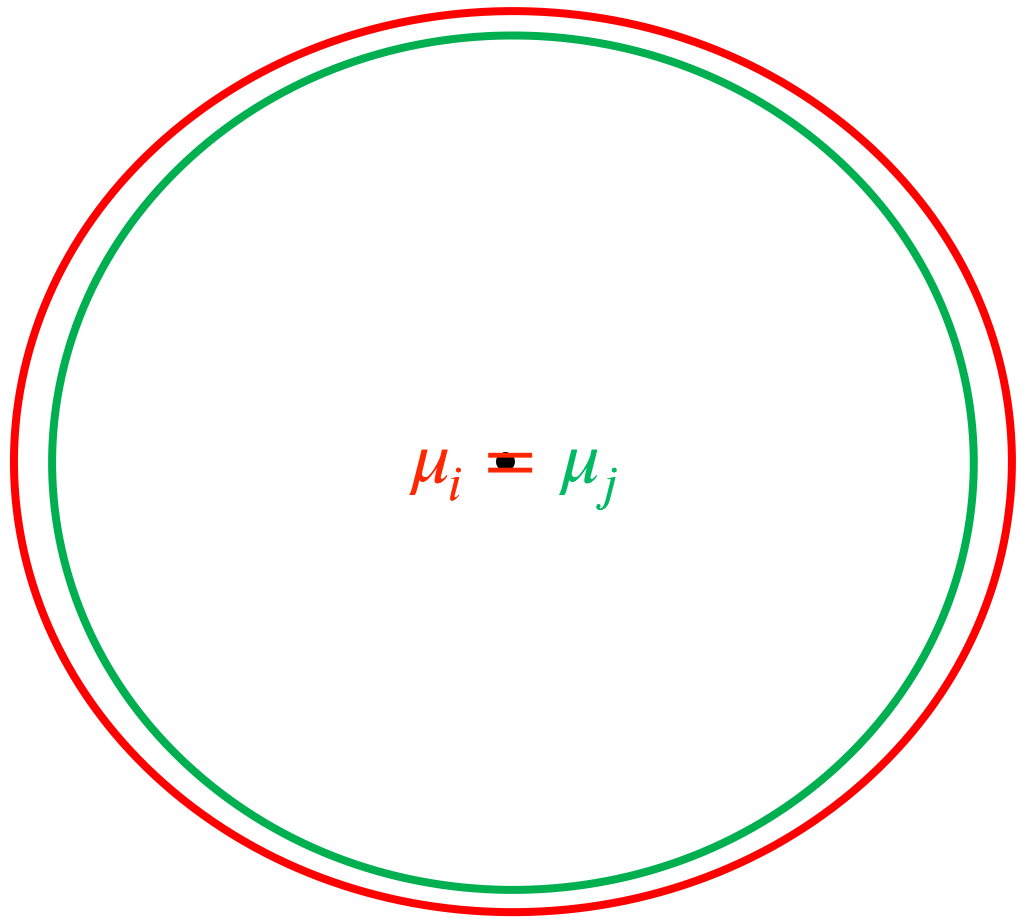
spectrally separated

$$\exists v: \frac{v^T \Sigma_i v}{v^T \Sigma_j v} > c$$



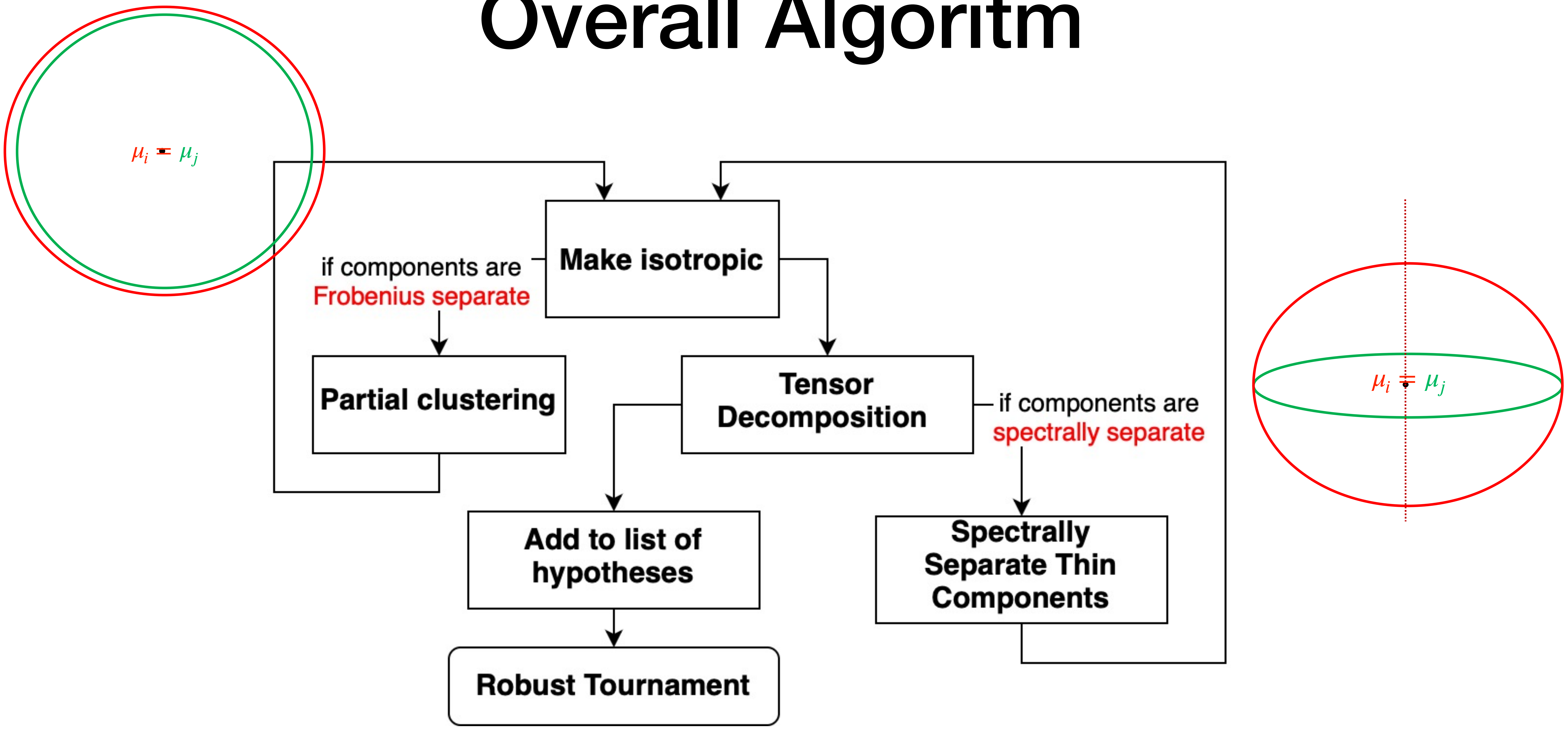
e.g., $\Sigma_i = I, \Sigma_j = I - vv^\top$

Frobenius separated

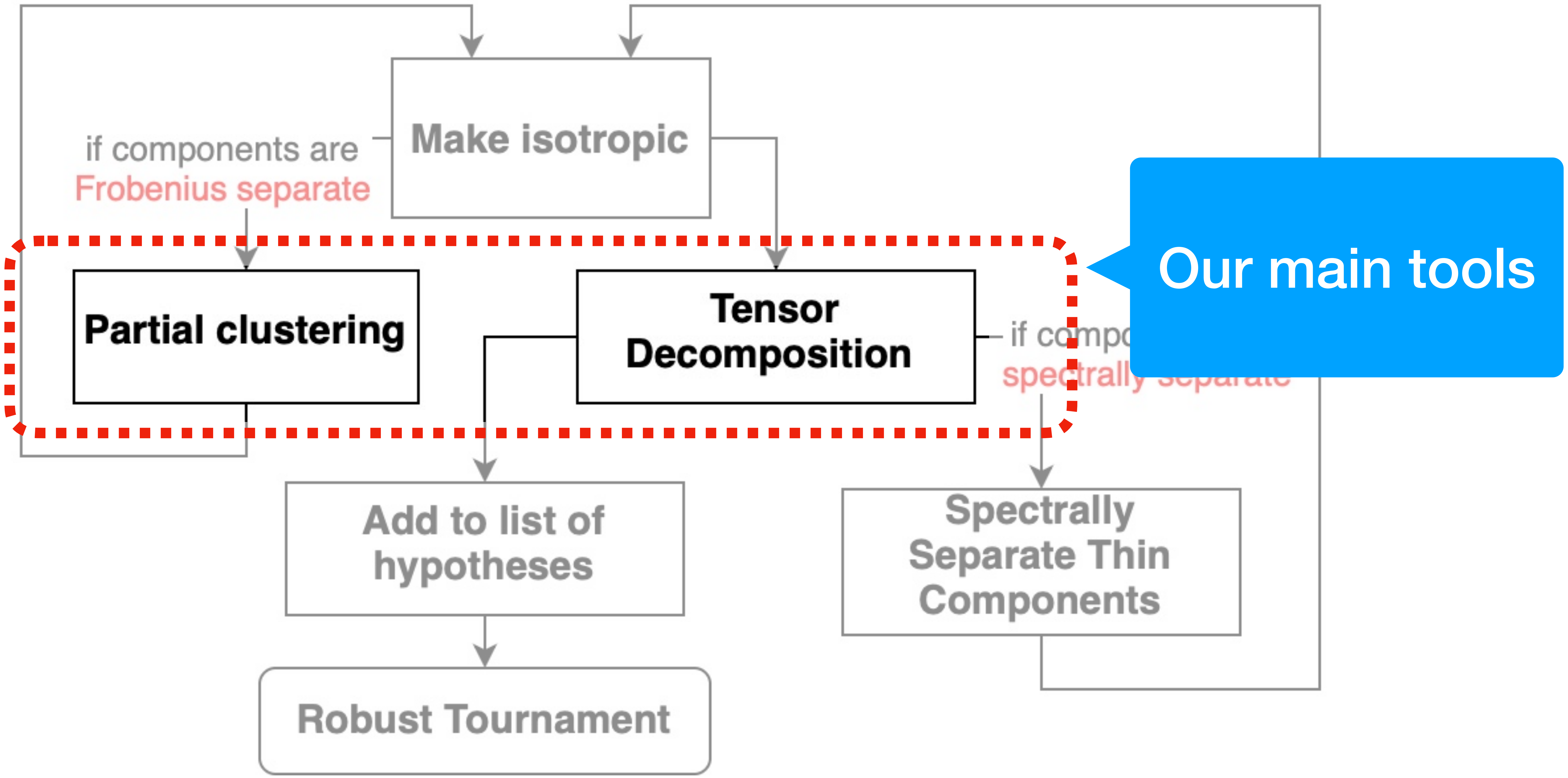


e.g., $\Sigma_i = I,\ \Sigma_j = \left(1 - \frac{100}{\sqrt{d}}\right) I$

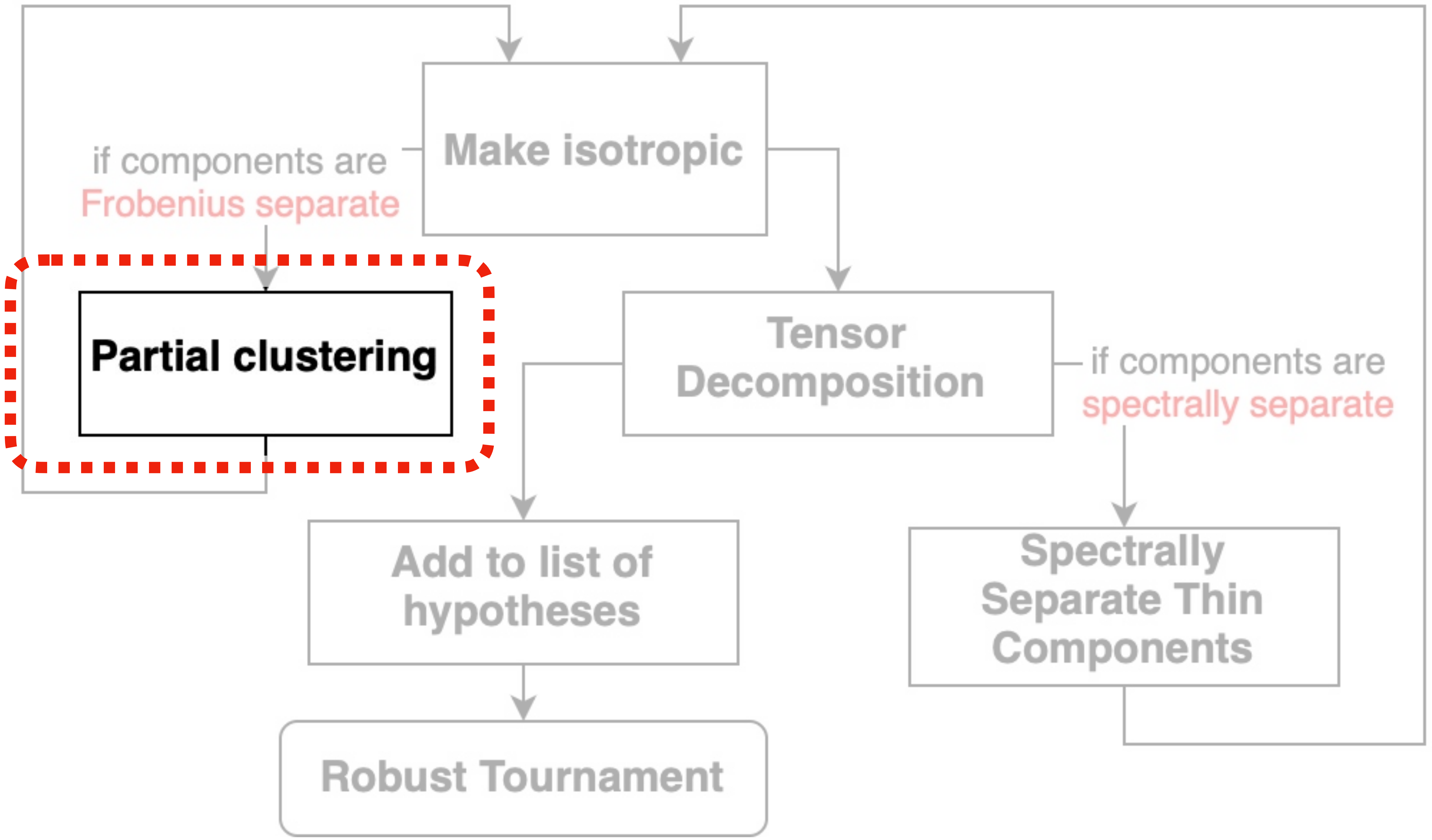
Overall Algoritm



Overall Algoritm

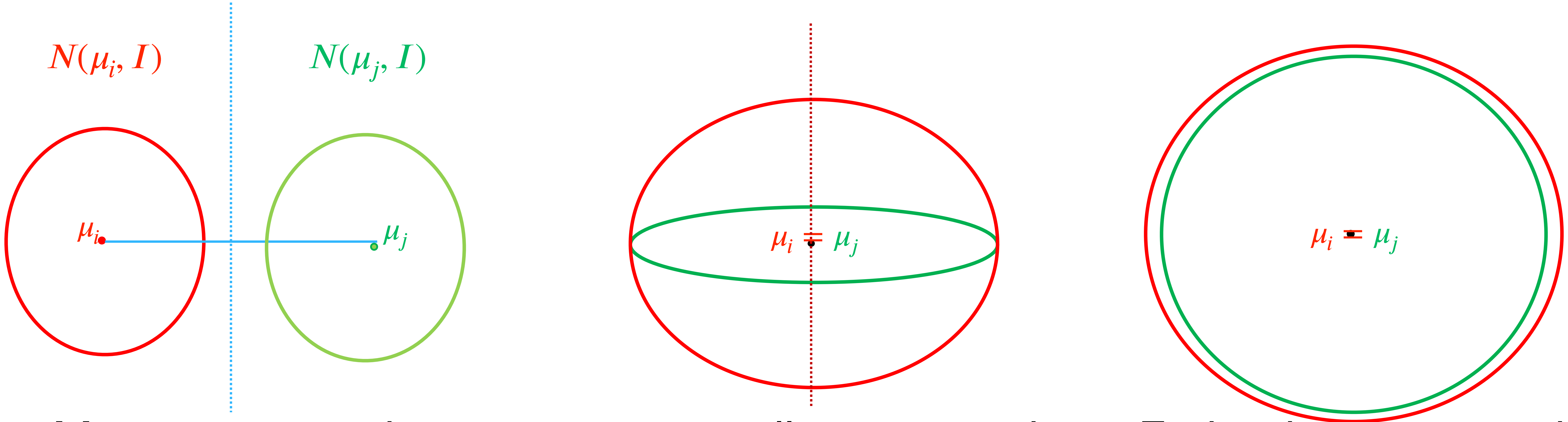


Step I: Cluster while you can



Robust Clustering

- [Bakshi-Kothari'20, Diakonikolas-Hopkins-Kane-Karmalkar'20]: robust clustering algorithms assuming the GMM is equiweighted ($w_i = 1/k$) and fully clusterable
  - Sum-of-Squares(SoS)-based clustering algorithm



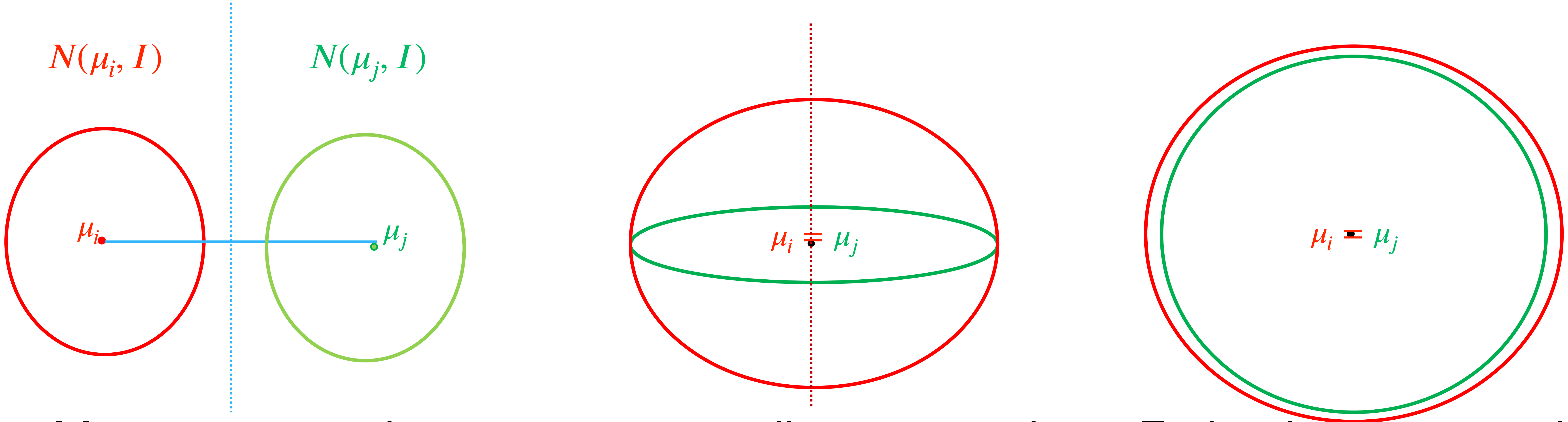
Mean-separated | spectrally separated | Frobenius separated

Robust Clustering

- [Bakshi-Kothari'20, Diakonikolas-Hopkins-Kane-Karmalkar'20]: robust clustering algorithms assuming the GMM is equiweighted ($w_i = 1/k$) and fully clusterable

  Each pair of components have TV distance at least $1 - \delta_k$

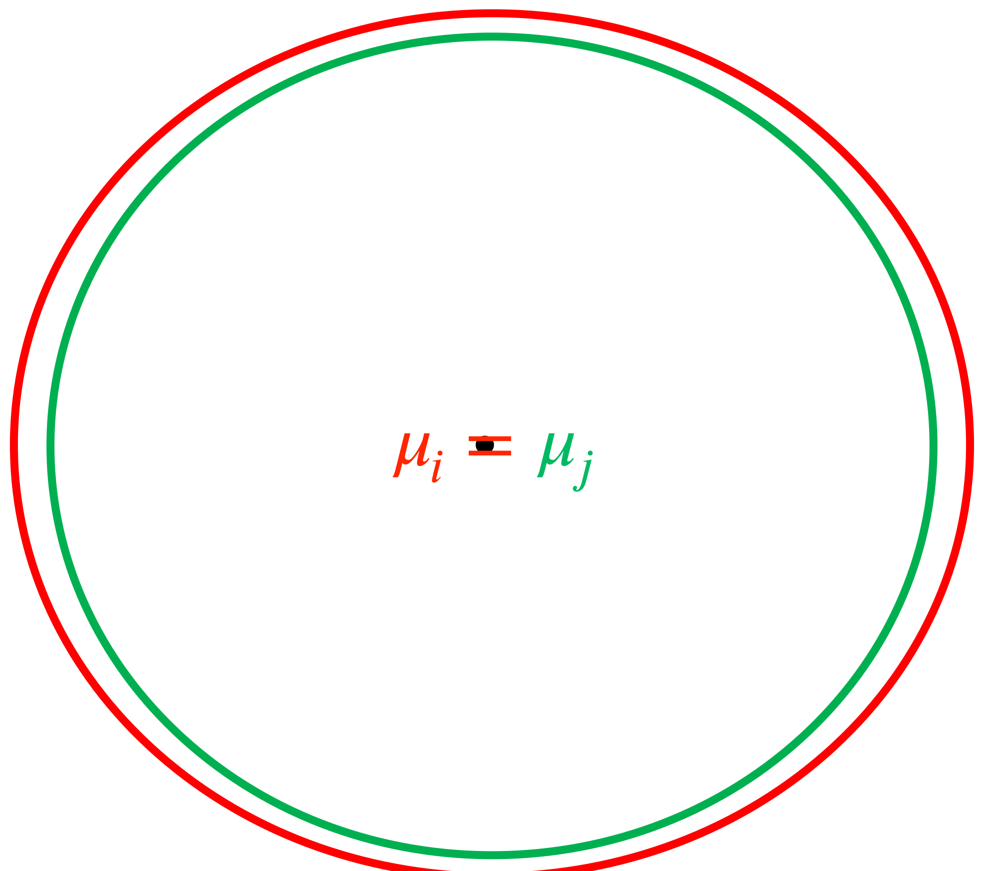
  - Sum-of-Squares(SoS)-based clustering algorithm



Mean-separated | spectrally separated | Frobenius separated

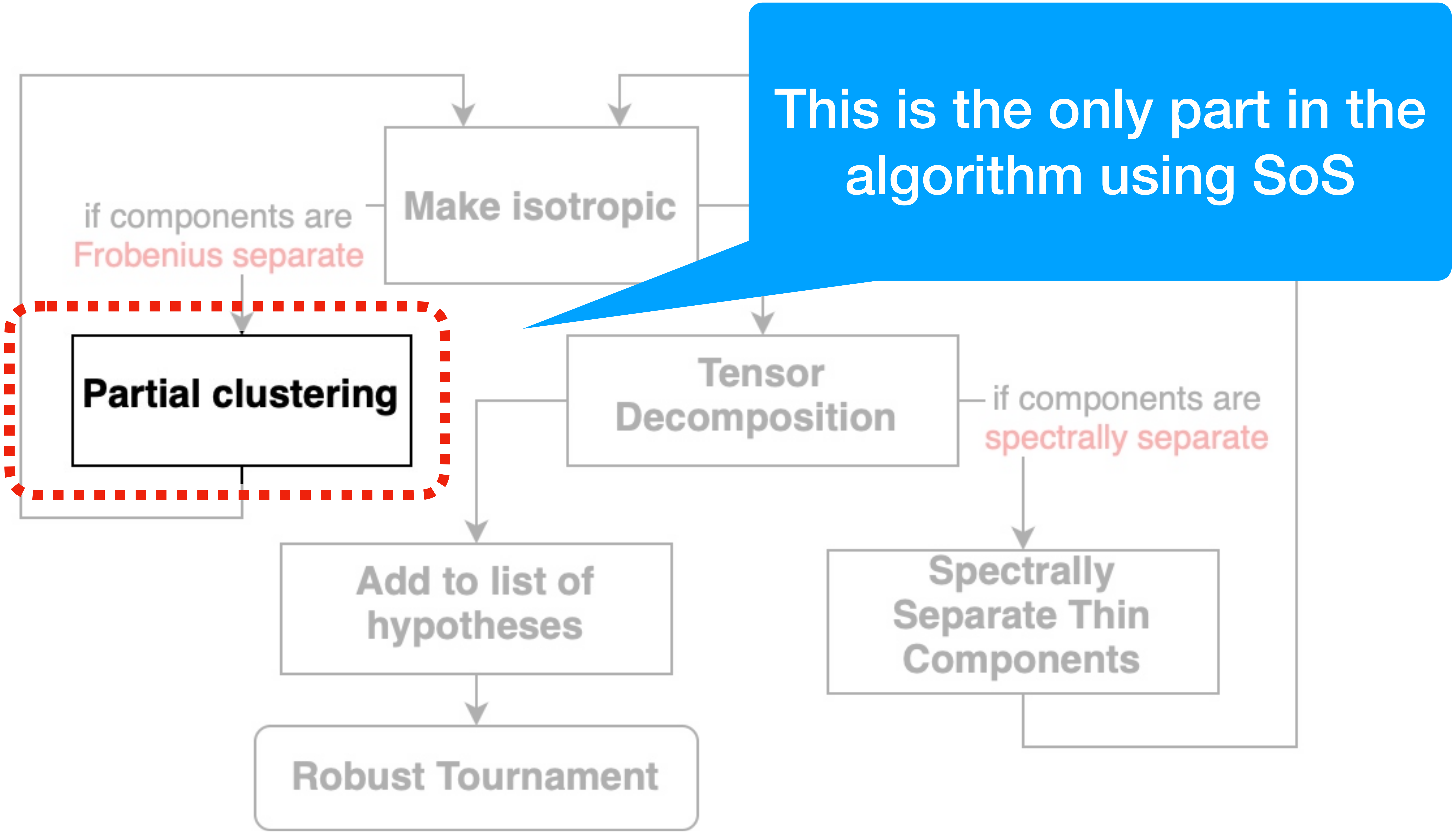
Robust Partial Clustering

- Using the basic SoS-based clustering algorithm of [BK20] gives a partial clustering algorithm with exponential dependence on $w_{\min}$, the minimum mixing weight.
- To avoid this, we only do partial clustering if there is separation in Frobenius norm.
- Theorem: Robust partial clustering assuming Frobenius separation takes only $d^{O(1)}\mathrm{poly}_k(1/\epsilon)$ time and samples.
  - Based on a new SoS relaxation and rounding



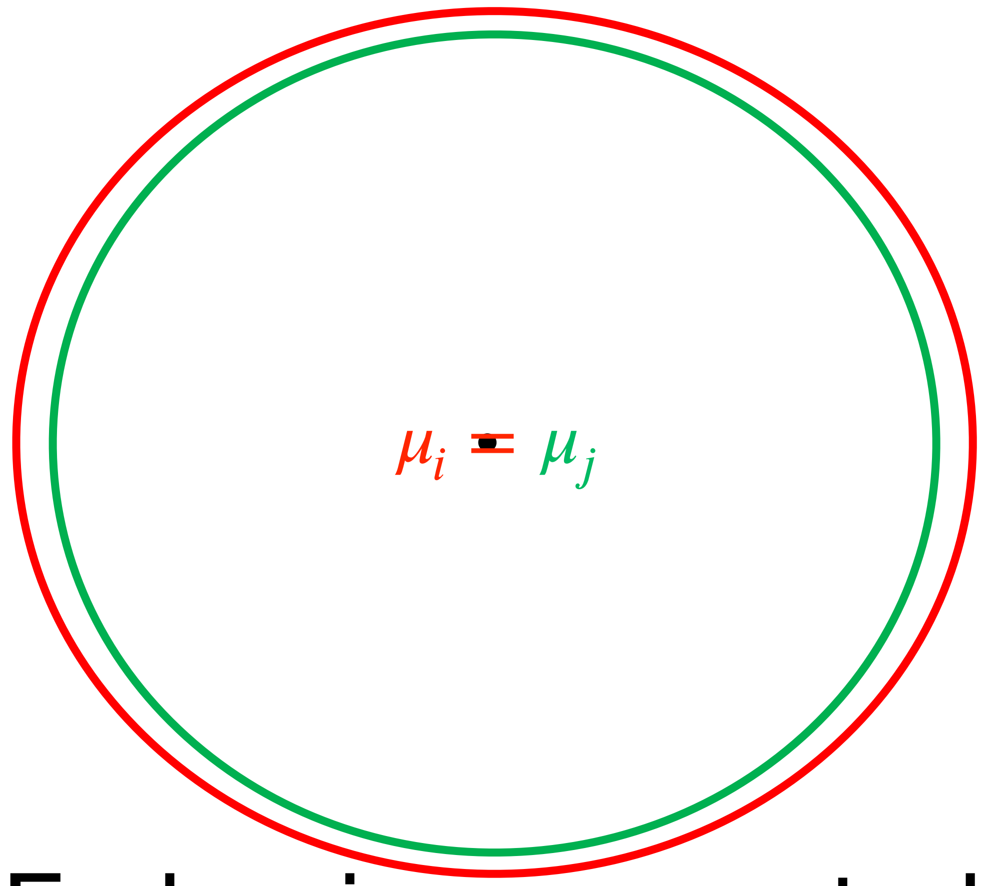
Frobenius separated

Robust Partial Clustering



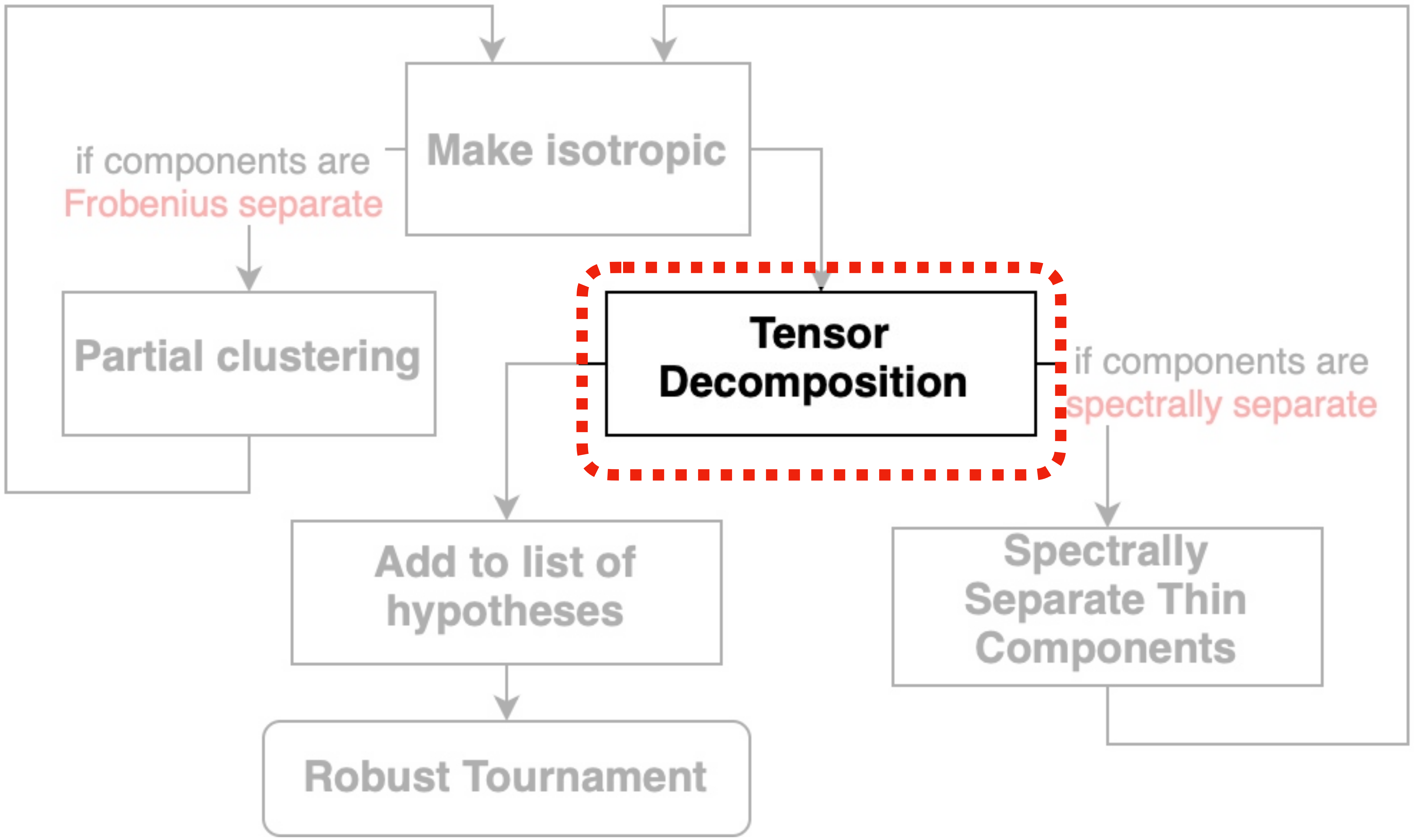
Non-SoS Robust Partial Clustering

- [Diakonikolas-Kane-Lee-Pensia-Pittas'23]: robust partial clustering algorithm assuming Frobenius separation
  - Spectral based "filtering"
  - SoS-free algorithm for robustly learning GMMs



Frobenius separated

Step II: Learn non-clusterable mixtures



Robust Tensor decomposition

- List recovery of means and covariances of each non-clusterable components with the assumption that
    - component covariances are close to identity

Robust Tensor decomposition

- List recovery of means and covariances of each non-clusterable components with the assumption that
  - component covariances are close to identity

Non-Frobenius separable

So we need to do partial clustering first!

Robust Tensor decomposition

A variant of moments, can be estimated efficiently

- “Method of moments”: Hermite tensors
  - [Kane’20]: An efficient algorithm for equiweighted mixtures of 2 Gaussians using Hermite tensors

Learning Covariances up to low-rank error

- Random collapsing the 4th Hermite tensor recovers the covariances with low-rank terms
- 4th Hermite Tensor $T_4 = \mathbb{E}[h_4(X)] = \mathrm{Sym}\left(\sum_{i=1}^{k} w_i(3S_i \otimes S_i + 6S_i \otimes \mu_i^{\otimes 2} + \mu_i^{\otimes 4})\right)$

$S_i = \Sigma_i - I$

- $S_i' = S_i + \mu_i^{\otimes 2}, T_4' = \sum_i w_i(S_i' \otimes S_i')$
- $T_4 = \mathrm{Sym}\left(\sum_{i=1}^{k} w_i(3S_i' \otimes S_i' - 2\mu_i^{\otimes 4})\right)$
- $T_4(\cdot,\cdot,x,y) = T_4'(\cdot,\cdot,x,y) + \sum_i w_i(S_i'x) \otimes (S_i'y) + \sum_i w_i(S_i'y) \otimes (S_i'x) + \sum_i w_i(-2\mu_i^{\otimes 2}\mu_i^T x \mu_i^T y)$

Random Collapsing

Learning Covariances up to low-rank error

- Random collapsing the 4th Hermite tensor recovers the covariances with low-rank terms
- 4th Hermite Tensor $T_4 = \mathbb{E}[h_4(X)] = \mathrm{Sym}\left(\sum_{i=1}^{k} w_i(3S_i \otimes S_i + 6S_i \otimes \mu_i^{\otimes 2} + \mu_i^{\otimes 4})\right)$
- $S_i' = S_i + \boxed{\mu_i^{\otimes 2}}, T_4' = \sum_i w_i(S_i' \otimes S_i')$
- $T_4 = \mathrm{Sym}\left(\sum_{i=1}^{k} w_i(3S_i' \otimes S_i' - 2\mu_i^{\otimes 4})\right)$
- $T_4(\cdot,\cdot,x,y) = T_4'(\cdot,\cdot,x,y) + \boxed{\sum_i w_i(S_i'x) \otimes (S_i'y) + \sum_i w_i(S_i'y) \otimes (S_i'x) + \sum_i w_i(-2\mu_i^{\otimes 2}\mu_i^T x \mu_i^T y)}$

Low-rank terms

Learning Covariances up to low-rank error

- $S_i = \Sigma_i - I$
- $S_i' = S_i +$ Low-rank term, $T_4' = \sum_i w_i (S_i' \otimes S_i')$
- $T_4(\cdot, \cdot, x, y) = T_4'(\cdot, \cdot, x, y) +$ Low-rank terms
- By collapsing T_4 multiple times along random vector pairs x, y, and taking random linear combinations, we get approximations to S_i, up to low rank and small norm error terms.

Recover the low-rank terms and means

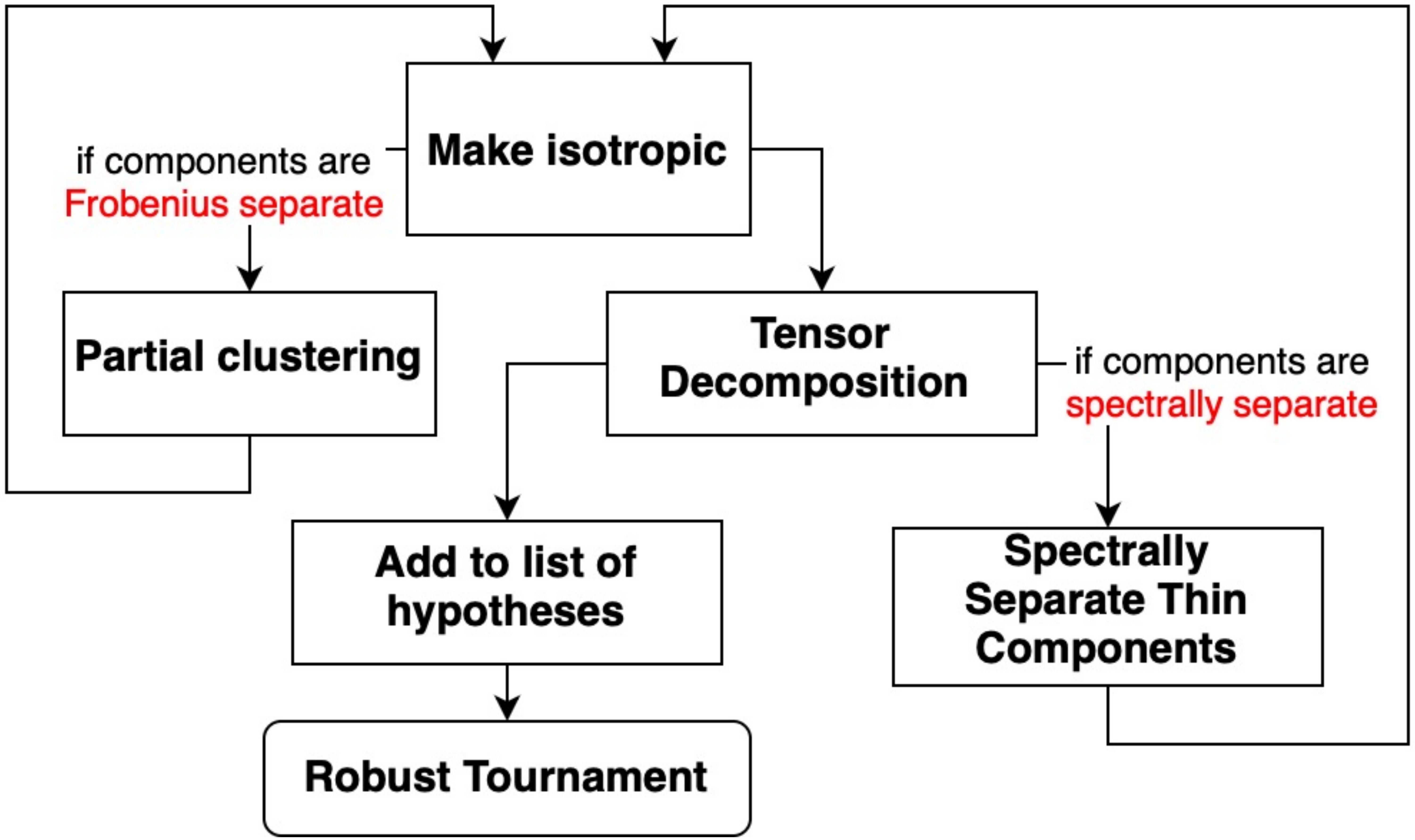
- μ_i and eigenvectors of S_i are in low-dimensional space

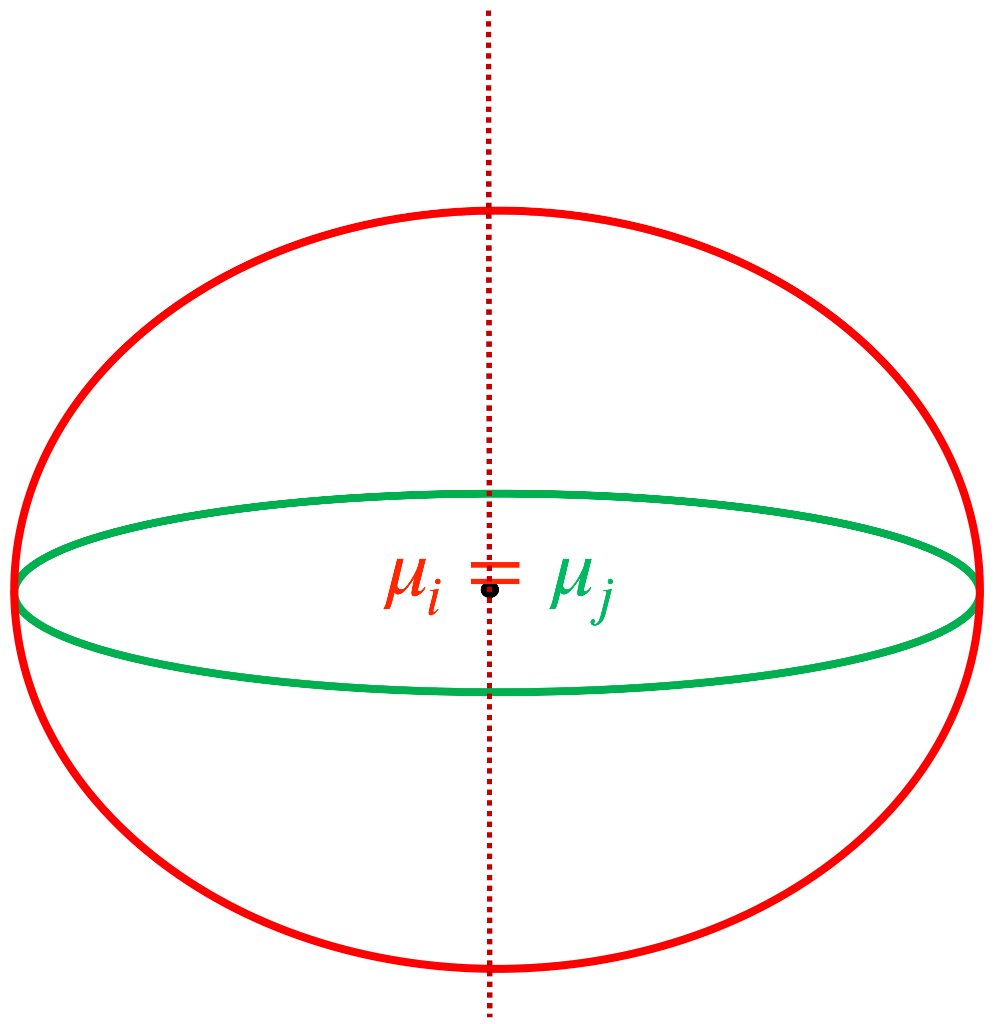
dim = $\mathrm{poly}_k(1/\epsilon)$

We can find the space by estimating the first $4k$ Hermite tensors

- Run [Moitra-Valiant'10] in low-dimensional space to recovery μ_i and S_i

Overall Algoritm





Thank you!