# Clustering Mixtures of Bounded Covariance Distributions Under Optimal Separation

Jasper Lee

University of Wisconsin-Madison

Joint work with Ilias Diakonikolas, Daniel Kane, Thanasis Pittas

# Clustering Mixture Distributions

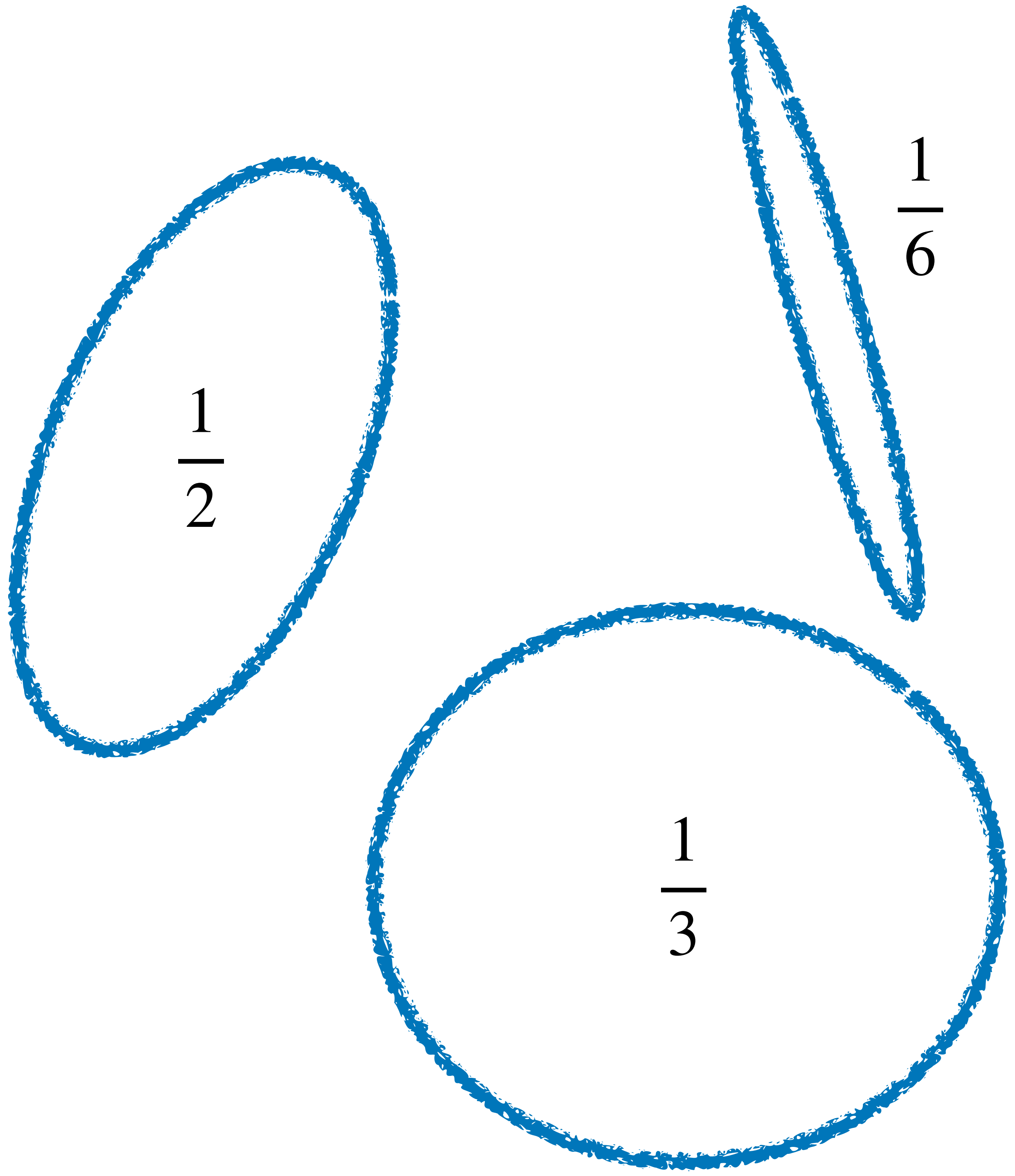
# Clustering Mixture Distributions

**Mixture model:**

# Clustering Mixture Distributions

**Mixture model:**

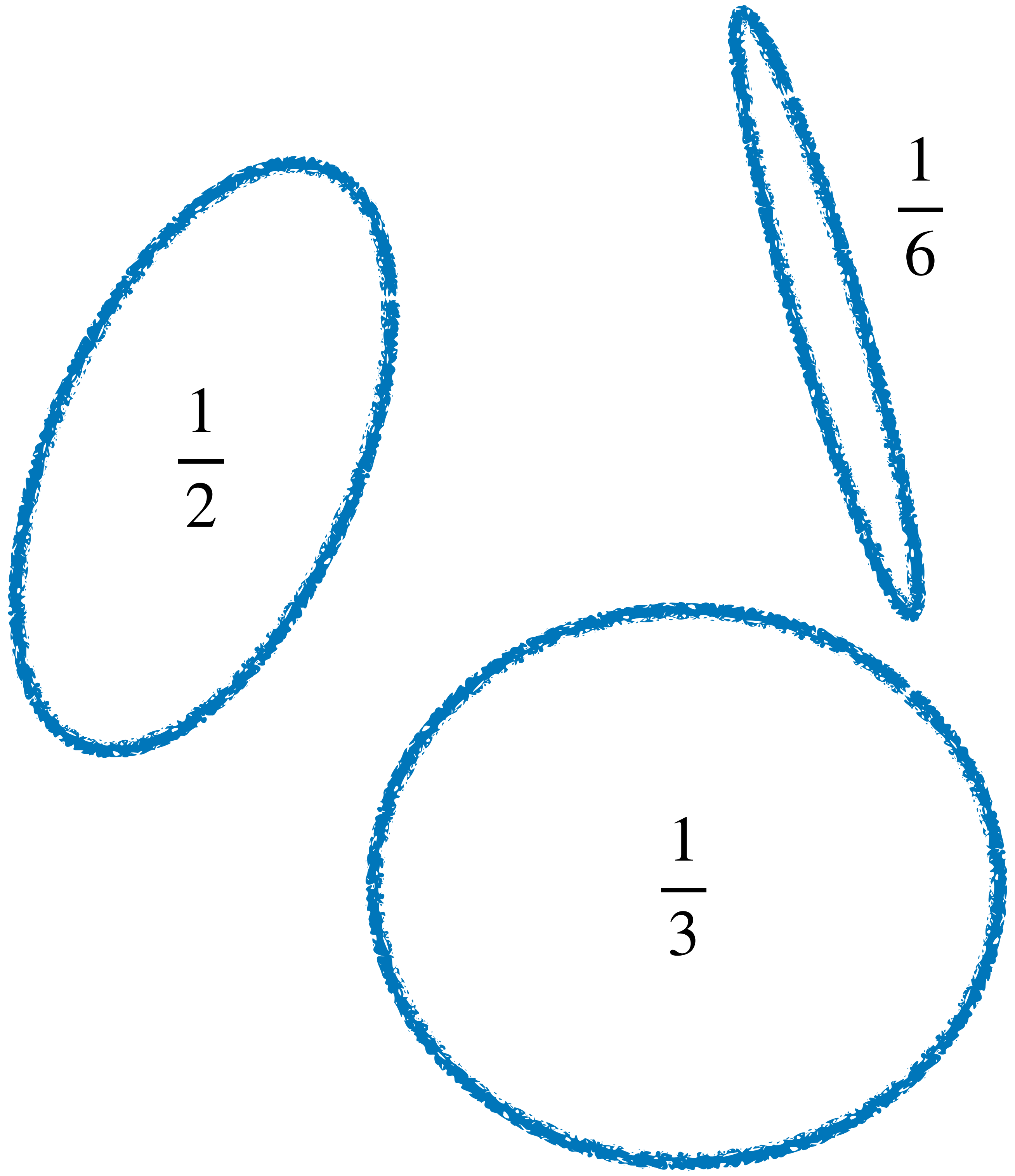
$$D = \frac{1}{2}P_1 + \frac{1}{3}P_2 + \frac{1}{6}P_3$$



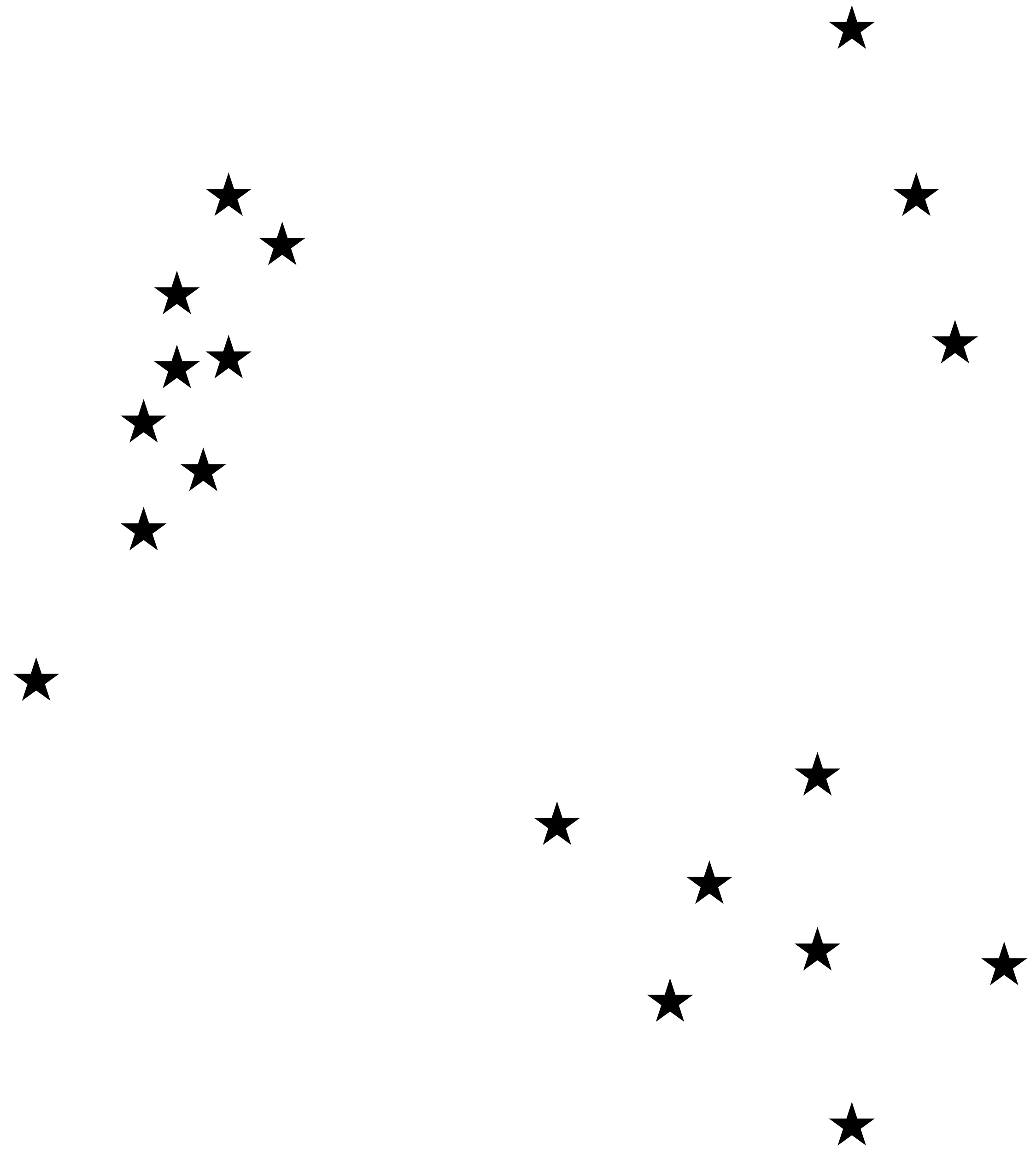
# Clustering Mixture Distributions

**Mixture model:**

$$D = \frac{1}{2}P_1 + \frac{1}{3}P_2 + \frac{1}{6}P_3$$



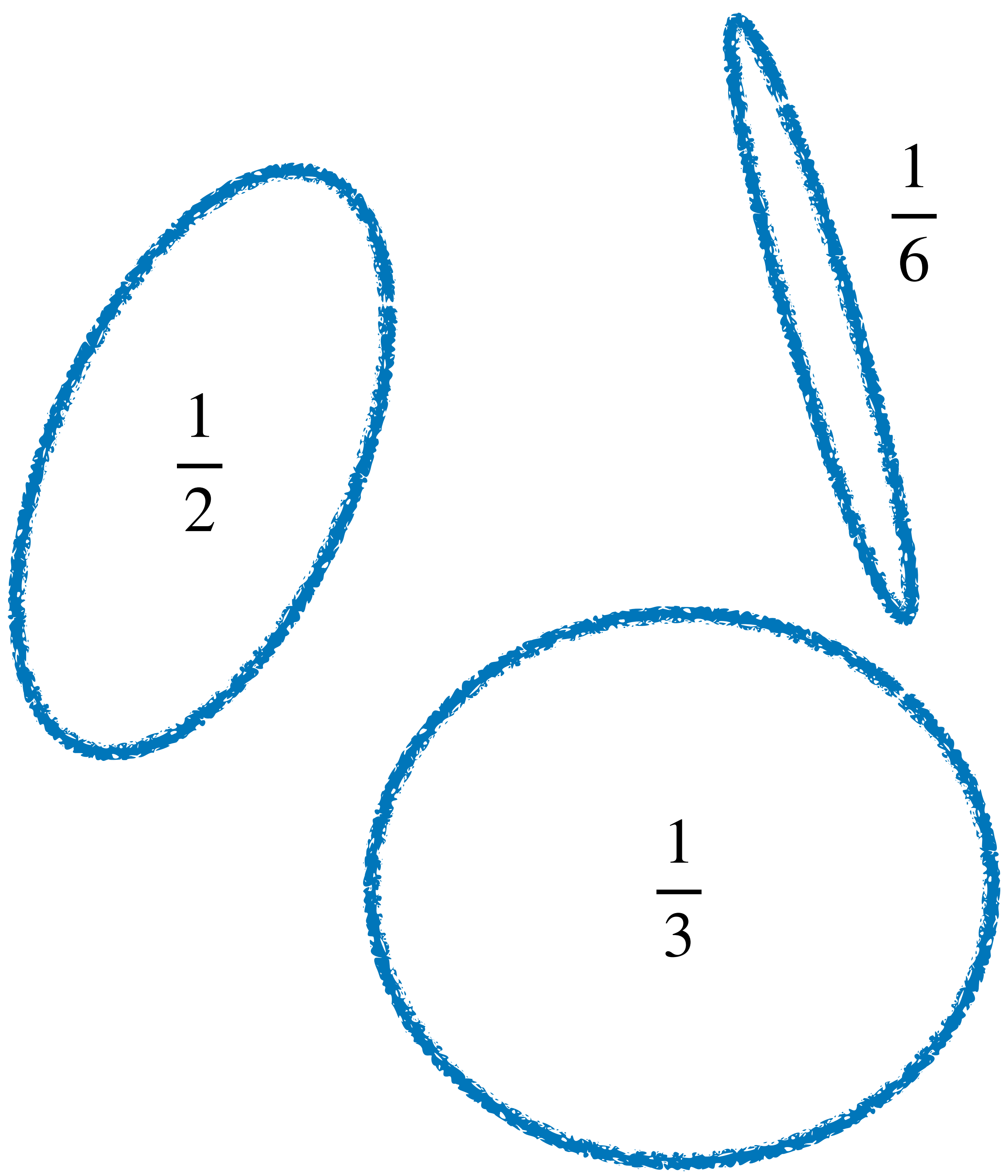

**Data:**



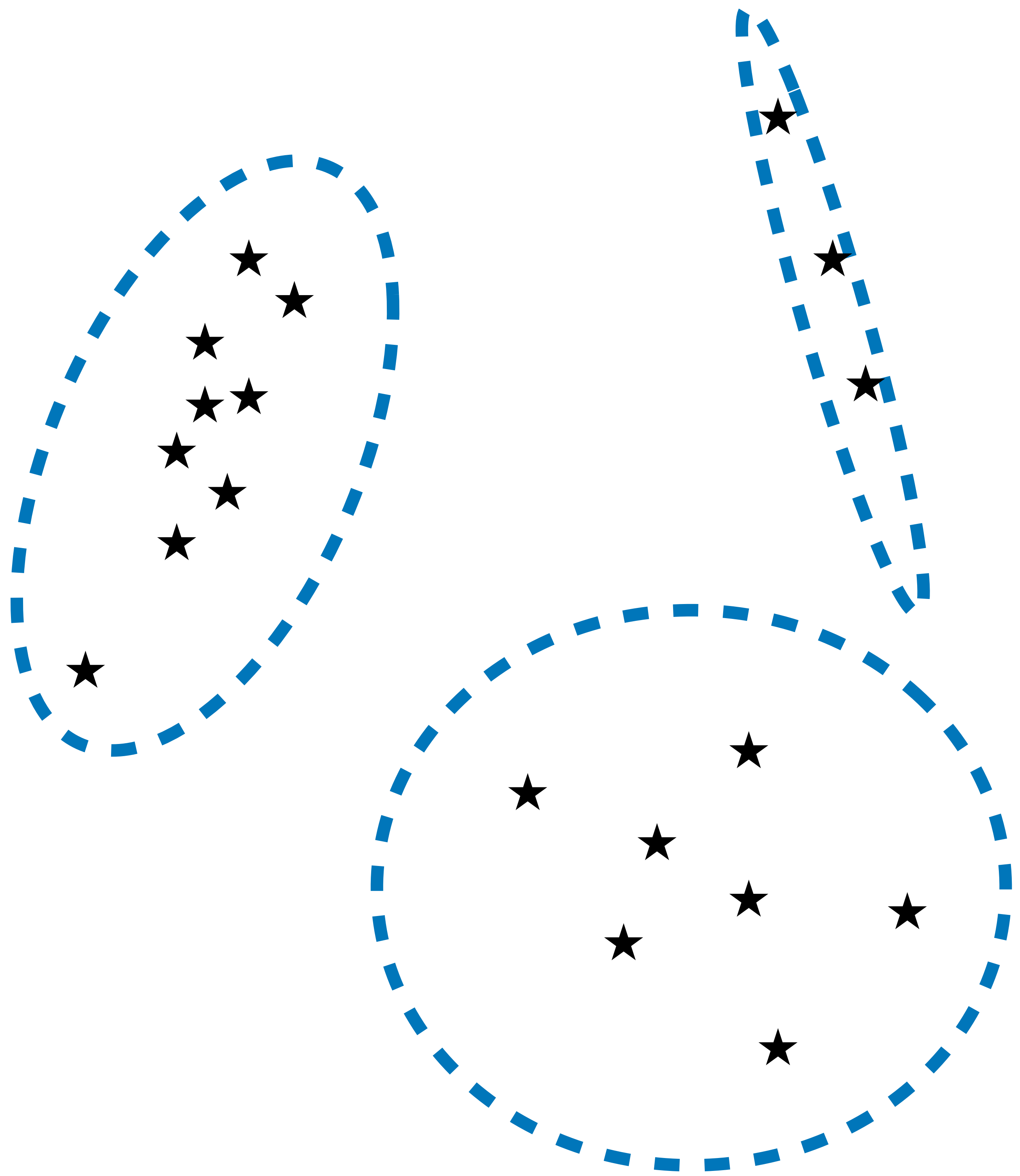
# Clustering Mixture Distributions

**Mixture model:**

$$D = \frac{1}{2}P_1 + \frac{1}{3}P_2 + \frac{1}{6}P_3$$




**Data:**



# Robust Clustering Mixture Distributions

arXiv:2312.11769

**Setup:**

- $\epsilon$-contaminated samples from $k$-mixture $D = \sum_{i=1}^{k} w_i P_i$
- $P_i$ has mean $\mu_i$ and covariance $\Sigma_i$, both unknown
- $\mu_i, \mu_j$ "well separated"

# Robust Clustering Mixture Distributions

arXiv:2312.11769

**Setup:**

- $\epsilon$-contaminated samples from $k$-mixture $D = \sum_{i=1}^{k} w_i P_i$
- $P_i$ has mean $\mu_i$ and covariance $\Sigma_i$, both unknown
- $\mu_i, \mu_j$ "well separated"

**Goal:** Identify 95% of the samples correctly for **every** cluster

# Robust Clustering Mixture Distributions

arXiv:2312.11769

$\epsilon \leq \min w_i / 100$

**Setup:**

- $\epsilon$-contaminated samples from $k$-mixture $D = \sum_{i=1}^{k} w_i P_i$
- $P_i$ has mean $\mu_i$ and covariance $\Sigma_i$, both unknown
- $\mu_i, \mu_j$ "well separated"

**Goal:** Identify 95% of the samples correctly for **every** cluster

# Robust Clustering Mixture Distributions

arXiv:2312.11769

$\epsilon \leq \min w_i / 100$

**Setup:**

- $\epsilon$-contaminated samples from $k$-mixture $D = \sum_{i=1}^{k} w_i P_i$
- $P_i$ has mean $\mu_i$ and covariance $\Sigma_i$, both unknown
- $\mu_i, \mu_j$ “well separated”

**What is the minimum separation?**

**Goal:** Identify 95% of the samples correctly for **every** cluster

# Minimum Separation - Uniform Mixtures

arXiv:2312.11769

# Minimum Separation - Uniform Mixtures

arXiv:2312.11769

**Assume**: Uniform mixture (for now)

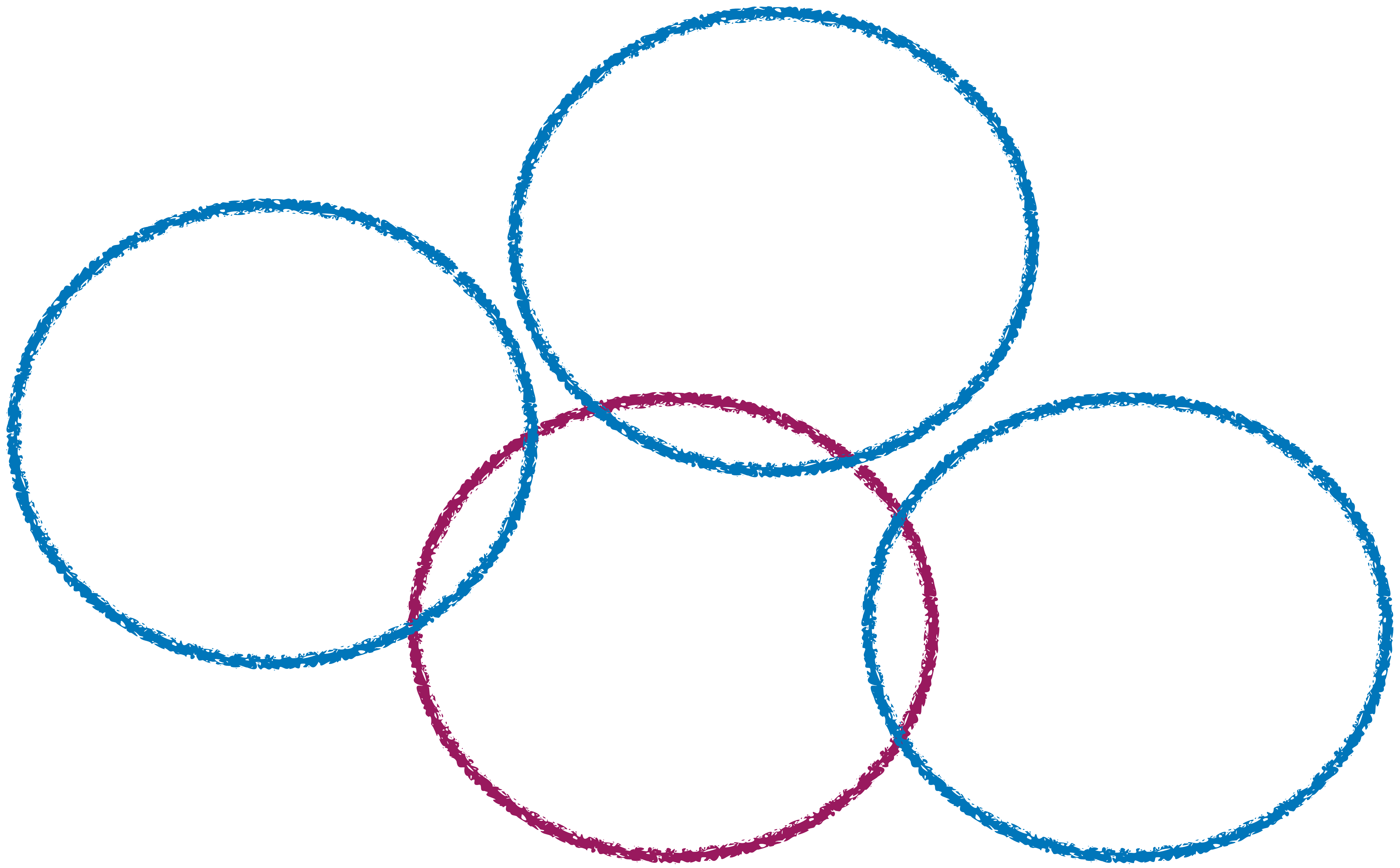
$$D = \sum_{i=1}^{k} \frac{1}{k} P_i$$

# Minimum Separation - Uniform Mixtures

arXiv:2312.11769

**Assume**: Uniform mixture (for now)

$$D = \sum_{i=1}^{k} \frac{1}{k} P_i$$



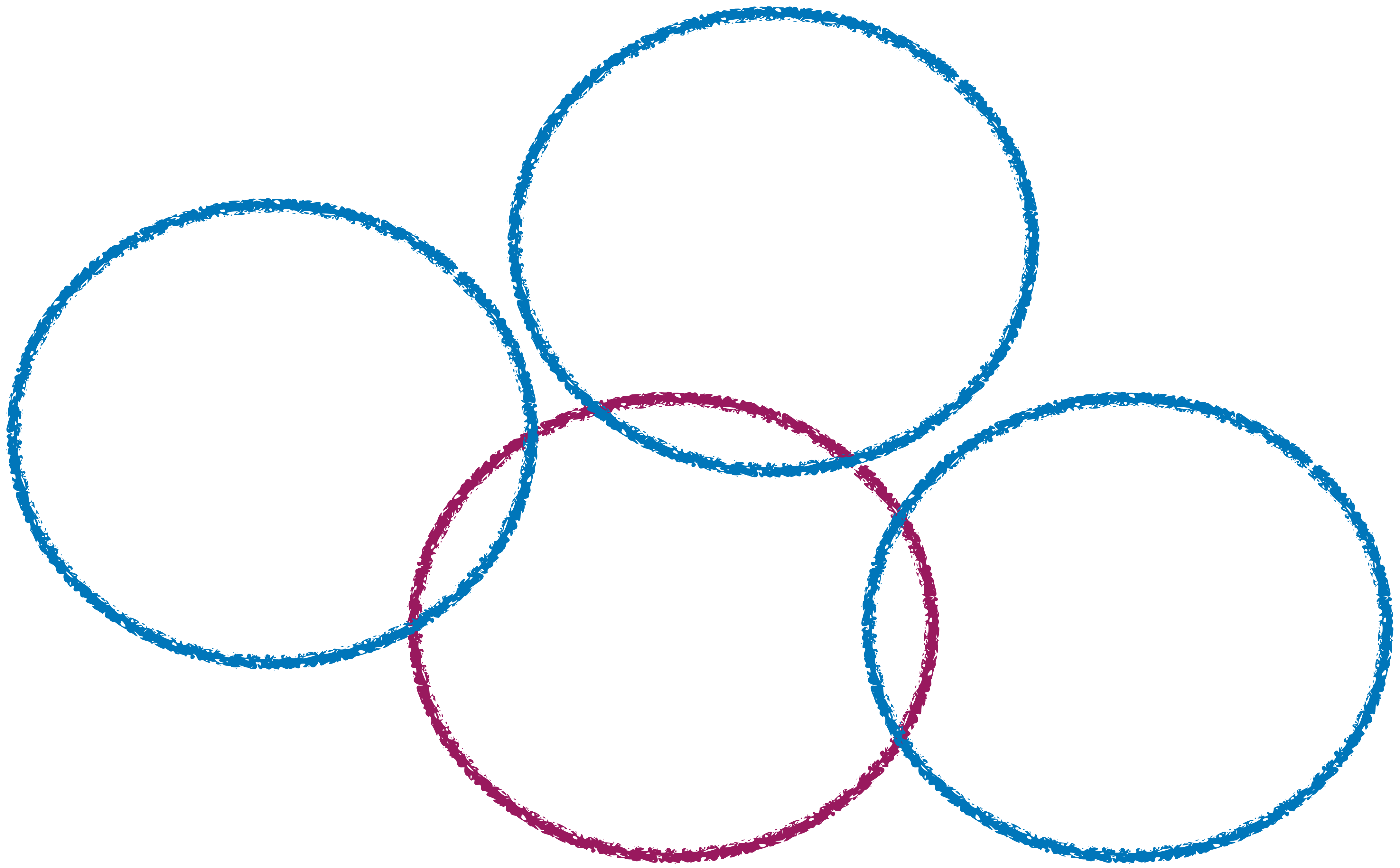
# Minimum Separation - Uniform Mixtures

arXiv:2312.11769

**Assume**: Uniform mixture (for now)

$$D = \sum_{i=1}^{k} \frac{1}{k} P_i$$

**Pairwise** overlap fraction $\lesssim 1/k$



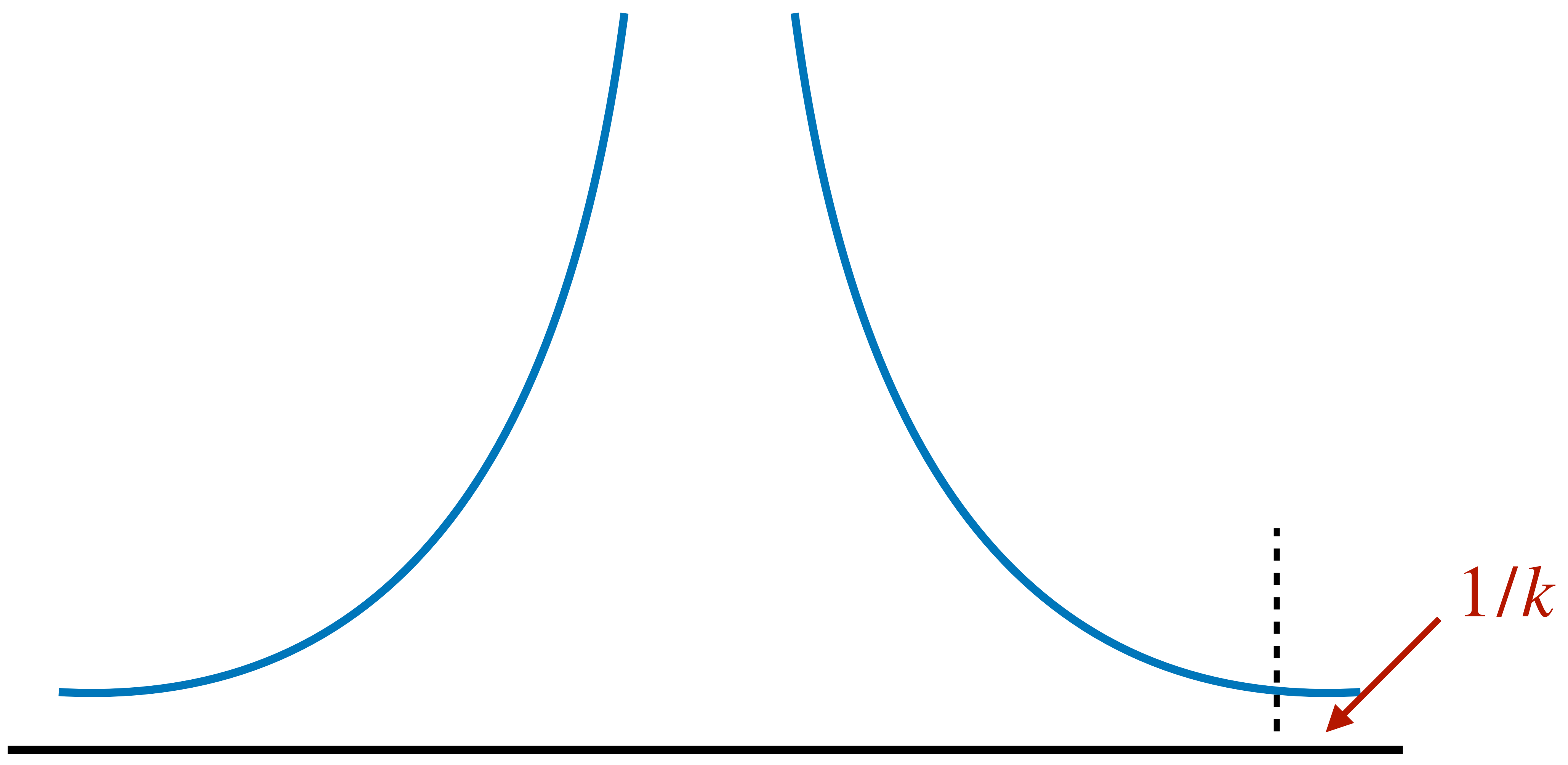
# Minimum Separation - Uniform Mixtures

arXiv:2312.11769

**Assume**: Uniform mixture (for now)

$$D = \sum_{i=1}^{k} \frac{1}{k} P_i$$

**Pairwise** overlap fraction $\lesssim 1/k$



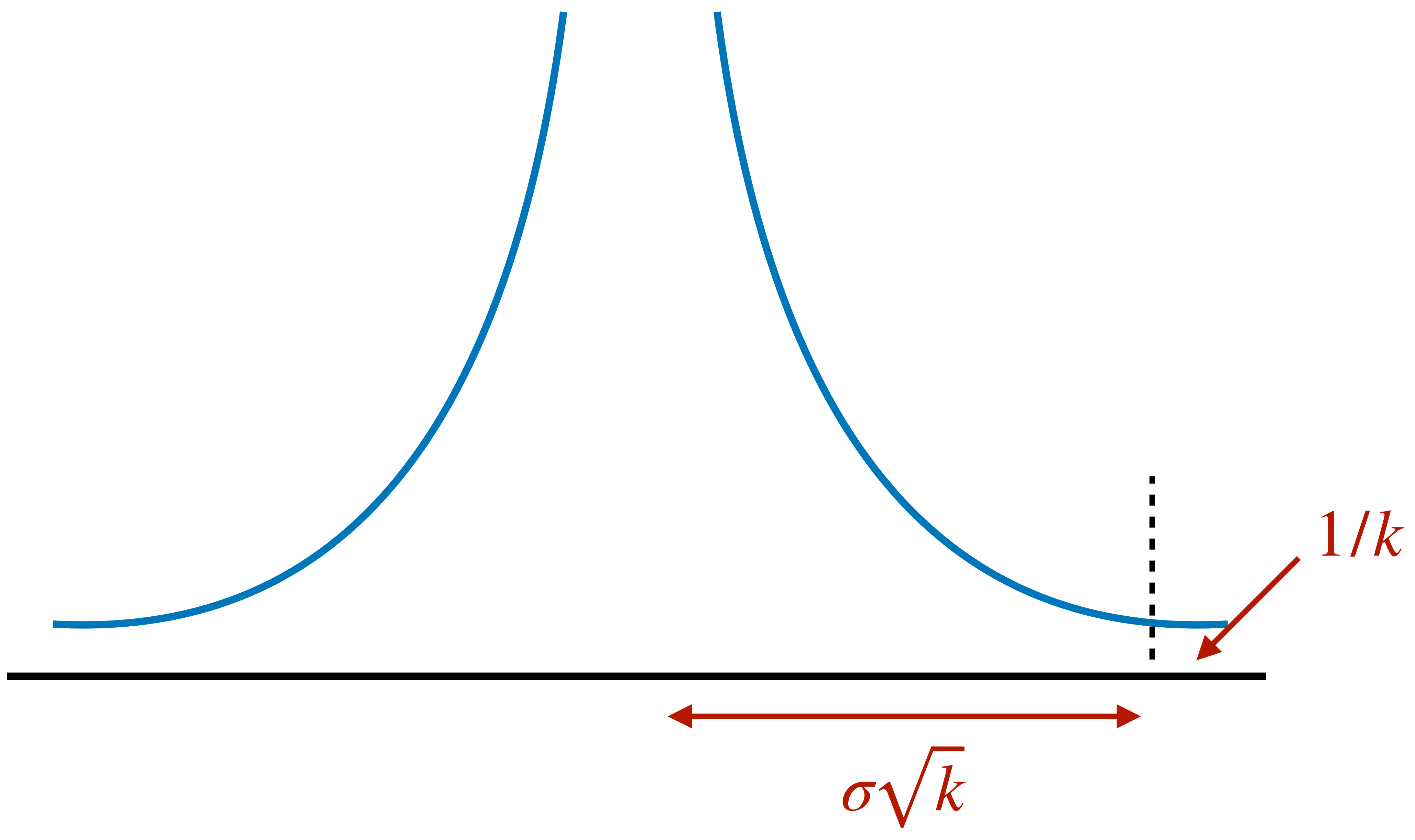
# Minimum Separation - Uniform Mixtures

arXiv:2312.11769

**Assume**: Uniform mixture (for now)

$$D = \sum_{i=1}^{k} \frac{1}{k} P_i$$

**Pairwise** overlap fraction $\lesssim 1/k$



# Minimum Separation - Uniform Mixtures

arXiv:2312.11769

**Assume**: Uniform mixture (for now)

$$D = \sum_{i=1}^{k} \frac{1}{k} P_i$$

**Pairwise** overlap fraction $\lesssim 1/k$

**Natural Goal:** If all cluster covariances $\preceq \sigma^2 I$, assume separation $\gg \sigma\sqrt{k}$

# Minimum Separation - Uniform Mixtures

arXiv:2312.11769

**Assume**: Uniform mixture (for now)

$$D = \sum_{i=1}^{k} \frac{1}{k} P_i$$

**Pairwise** overlap fraction $\lesssim 1/k$

**Natural Goal:** If all cluster covariances $\preceq \sigma^2 I$, assume separation $\gg \sigma\sqrt{k}$

**Solved! [DKKLT22]**: Near-linear time algorithm, also for list-decodable mean estimation

# Minimum Separation - Uniform Mixtures

arXiv:2312.11769

**Natural Goal:** If all cluster covariances $\preceq \sigma^2 I$, assume separation $\gg \sigma\sqrt{k}$

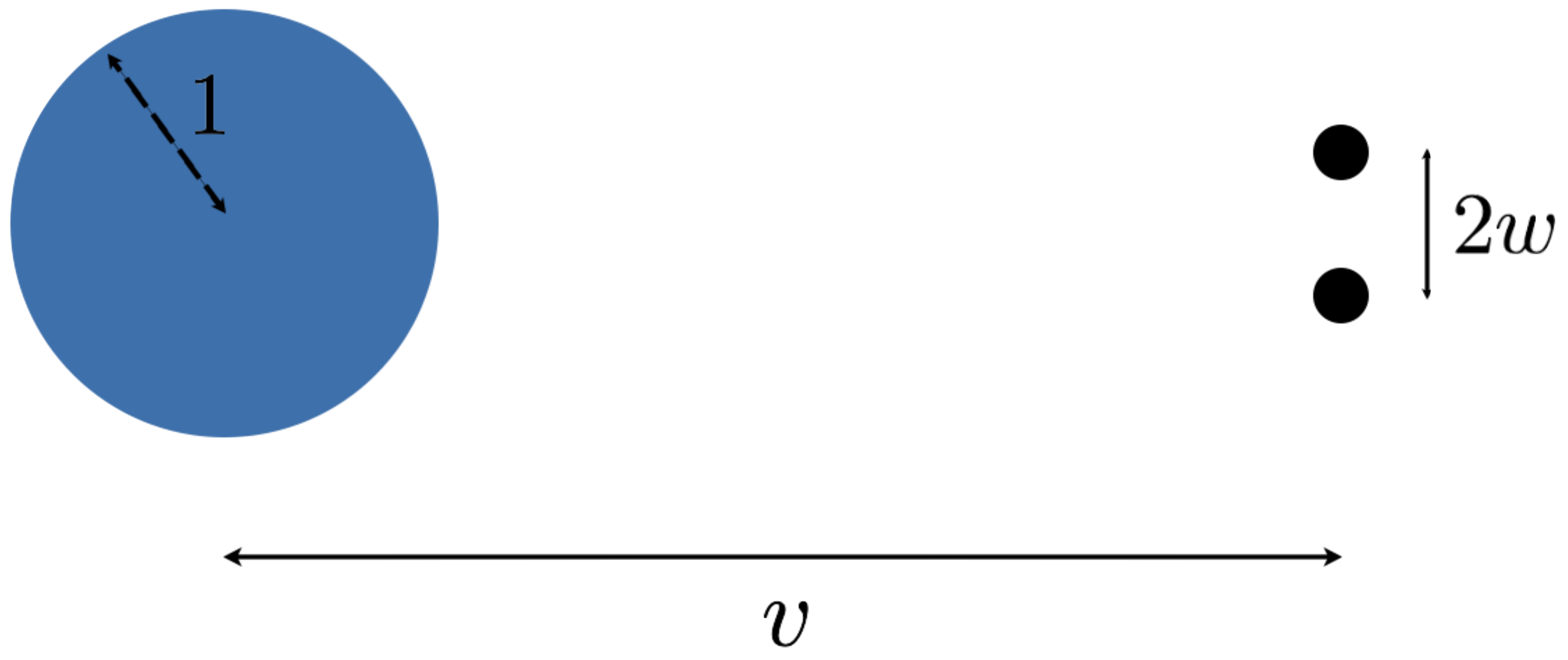
**Solved! [DKKLT22]**: Near-linear time algorithm, also for list-decodable mean estimation

# Minimum Separation - Uniform Mixtures

arXiv:2312.11769

**Natural Goal:** If all cluster covariances $\preceq \sigma^2 I$, assume separation $\gg \sigma\sqrt{k}$

**Solved! [DKKLT22]**: Near-linear time algorithm, also for list-decodable mean estimation

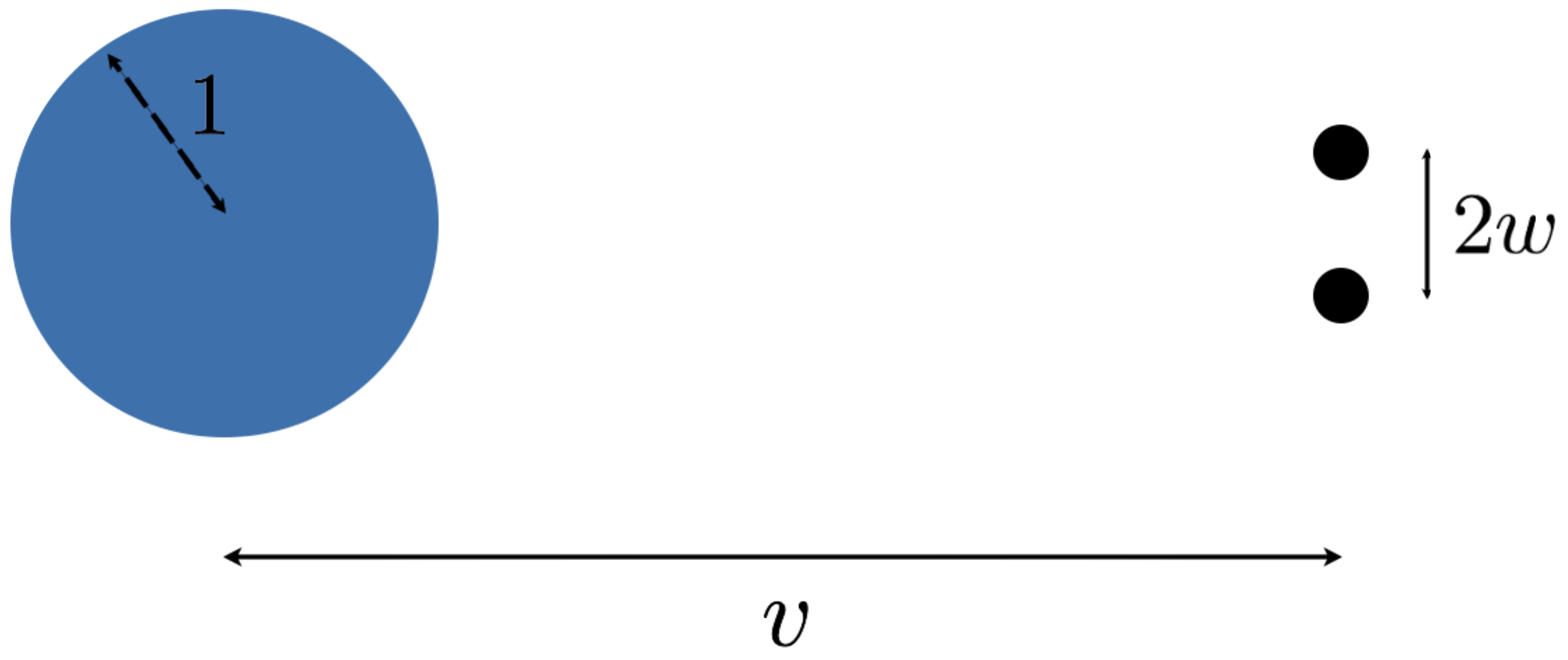


# Minimum Separation - Uniform Mixtures

arXiv:2312.11769

**Natural Goal:** If all cluster covariances $\preceq \sigma^2 I$, assume separation $\gg \sigma\sqrt{k}$

**Solved! [DKKLT22]**: Near-linear time algorithm, also for list-decodable mean estimation




**Clearly separable/clusterable**

but [DKKLT22] fails

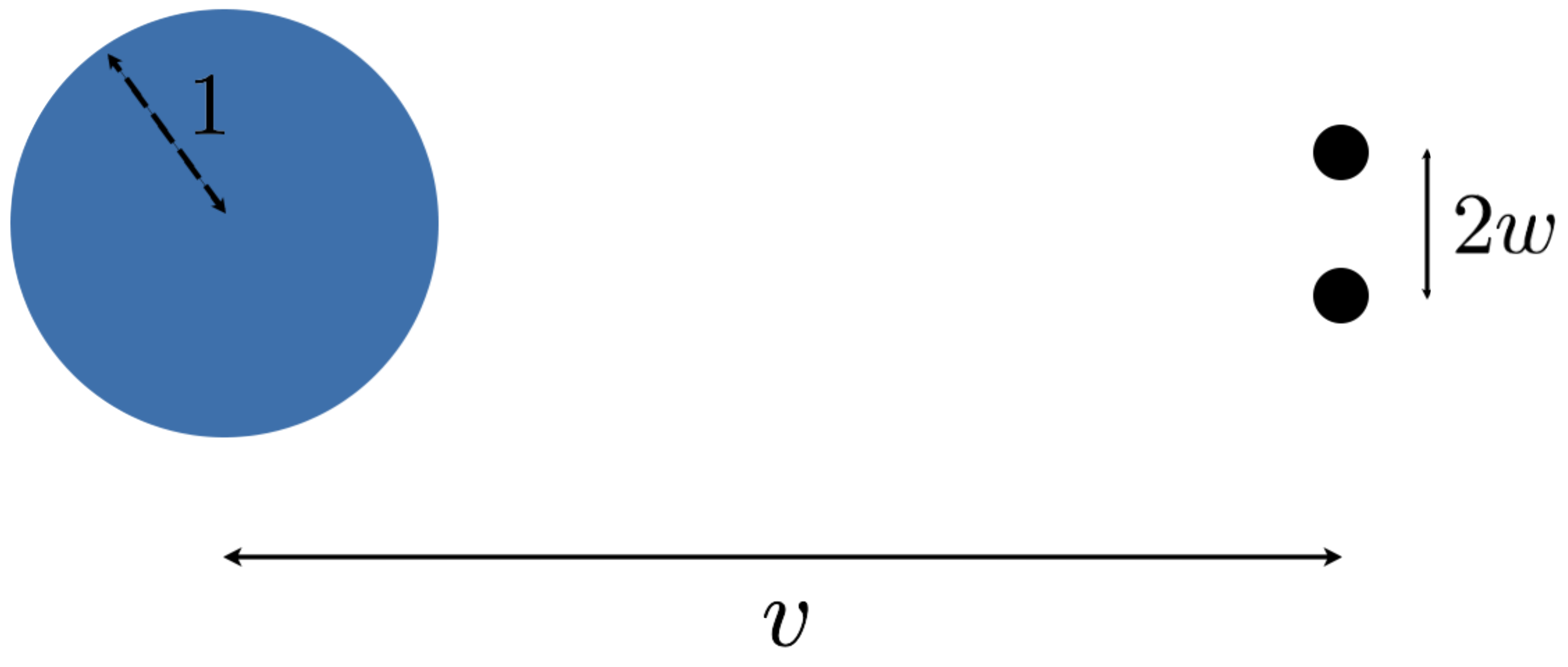
# Minimum Separation - Uniform Mixtures

arXiv:2312.11769

Fine-grained separation

**New Goal:** Between clusters $i, j$, assume separation $\gg (\sigma_i + \sigma_j)\sqrt{k}$

**Solved! [DKKLT22]**: Near-linear time algorithm, also for list-decodable mean estimation




**Clearly separable/clusterable**

but [DKKLT22] fails

# Minimum Separation - Uniform Mixtures

arXiv:2312.11769

Fine-grained separation

**New Goal:** Between clusters $i, j$, assume separation $\gg (\sigma_i + \sigma_j)\sqrt{k}$

# Minimum Separation - Uniform Mixtures

arXiv:2312.11769

Fine-grained separation

**New Goal:** Between clusters $i, j$, assume separation $\gg (\sigma_i + \sigma_j)\sqrt{k}$

**Prior work**:

- [DKKLT22]: $\max_i \sigma_i\sqrt{k}$ separation
- [BKK22]: $(\sigma_i + \sigma_j)\,\mathrm{poly}(k, \log n)$ separation + “No large sub-cluster” assumption

# Minimum Separation - Uniform Mixtures

arXiv:2312.11769

Fine-grained separation

**New Goal:** Between clusters $i, j$, assume separation $\gg (\sigma_i + \sigma_j)\sqrt{k}$

**Prior work**:

- [DKKLT22]: $\max_i \sigma_i \sqrt{k}$ separation
- [BKK22]: $(\sigma_i + \sigma_j)\,\mathrm{poly}(k, \log n)$ separation + "No large sub-cluster" assumption

**Our work**:

- [DKLP23]: $(\sigma_i + \sigma_j)\sqrt{k}$ separation

# Theorem — Uniform Mixtures

arXiv:2312.11769

Failure probability

Corrupted, $\epsilon \leq 1/(100k)$

**Theorem:** Given $\tilde{O}((d + \log 1/\delta)\, k^2)$ samples from $D = \sum_i \frac{1}{k} P_i$

where $P_i$ has mean $\mu_i$ and covariance $\Sigma_i \preceq \sigma_i^2 I$ (all unknown)

and $\|\mu_i - \mu_j\| \gg (\sigma_i + \sigma_j)\sqrt{k}$

***Algorithm*** returns sets $B_1, \ldots, B_k$ such that up to index permutation:

- $B_i$ overlaps with 95% of cluster $i$ samples $S_i$
- Mean of $B_i$ is $O(\sigma_i)$ close to $\mu_i$

# Theorem — Uniform Mixtures

arXiv:2312.11769

Failure probability

Corrupted, $\epsilon \leq 1/(100k)$

**Theorem:** Given $\tilde{O}((d + \log 1/\delta)\, k^2)$ samples from $D = \sum_i \frac{1}{k} P_i$

where $P_i$ has mean $\mu_i$ and covariance $\Sigma_i \preceq \sigma_i^2 I$ (all unknown)

and $\|\mu_i - \mu_j\| \gg (\sigma_i + \sigma_j)\sqrt{k}$

***Algorithm*** returns sets $B_1, \ldots, B_k$ such that up to index permutation:

- $B_i$ overlaps with 95% of cluster $i$ samples $S_i$
- Mean of $B_i$ is $O(\sigma_i)$ close to $\mu_i$

**Remarks**:

# Theorem — Uniform Mixtures

**arXiv:2312.11769**

**Failure probability** **Corrupted, $\epsilon \leq 1/(100k)$**

**Theorem:** Given $\tilde{O}((d + \log 1/\delta)\, k^2)$ samples from $D = \sum_i \frac{1}{k} P_i$

where $P_i$ has mean $\mu_i$ and covariance $\Sigma_i \preceq \sigma_i^2 I$ (all unknown)

and $\|\mu_i - \mu_j\| \gg (\sigma_i + \sigma_j)\sqrt{k}$

***Algorithm*** returns sets $B_1, \ldots, B_k$ such that up to index permutation:

- $B_i$ overlaps with 95% of cluster $i$ samples $S_i$
- Mean of $B_i$ is $O(\sigma_i)$ close to $\mu_i$

**Remarks**:

- Does not need to know $k$ precisely, only need input $\alpha \in [0.6/k, 1/k]$

# Theorem — Uniform Mixtures

arXiv:2312.11769

Failure probability

Corrupted, $\epsilon \leq 1/(100k)$

**Theorem:** Given $\tilde{O}((d + \log 1/\delta)\, k^2)$ samples from $D = \sum_i \frac{1}{k} P_i$

where $P_i$ has mean $\mu_i$ and covariance $\Sigma_i \preceq \sigma_i^2 I$ (all unknown)

and $\|\mu_i - \mu_j\| \gg (\sigma_i + \sigma_j)\sqrt{k}$

***Algorithm*** returns sets $B_1, \ldots, B_k$ such that up to index permutation:

- $B_i$ overlaps with 95% of cluster $i$ samples $S_i$
- Mean of $B_i$ is $O(\sigma_i)$ close to $\mu_i$

**Remarks**:

- Does not need to know $k$ precisely, only need input $\alpha \in [0.6/k, 1/k]$
- Can work for almost-uniform mixtures, with each $w_i \in [0.9/k, 1.1/k]$

# Algorithm — Uniform Mixtures

arXiv:2312.11769

# Algorithm — Uniform Mixtures

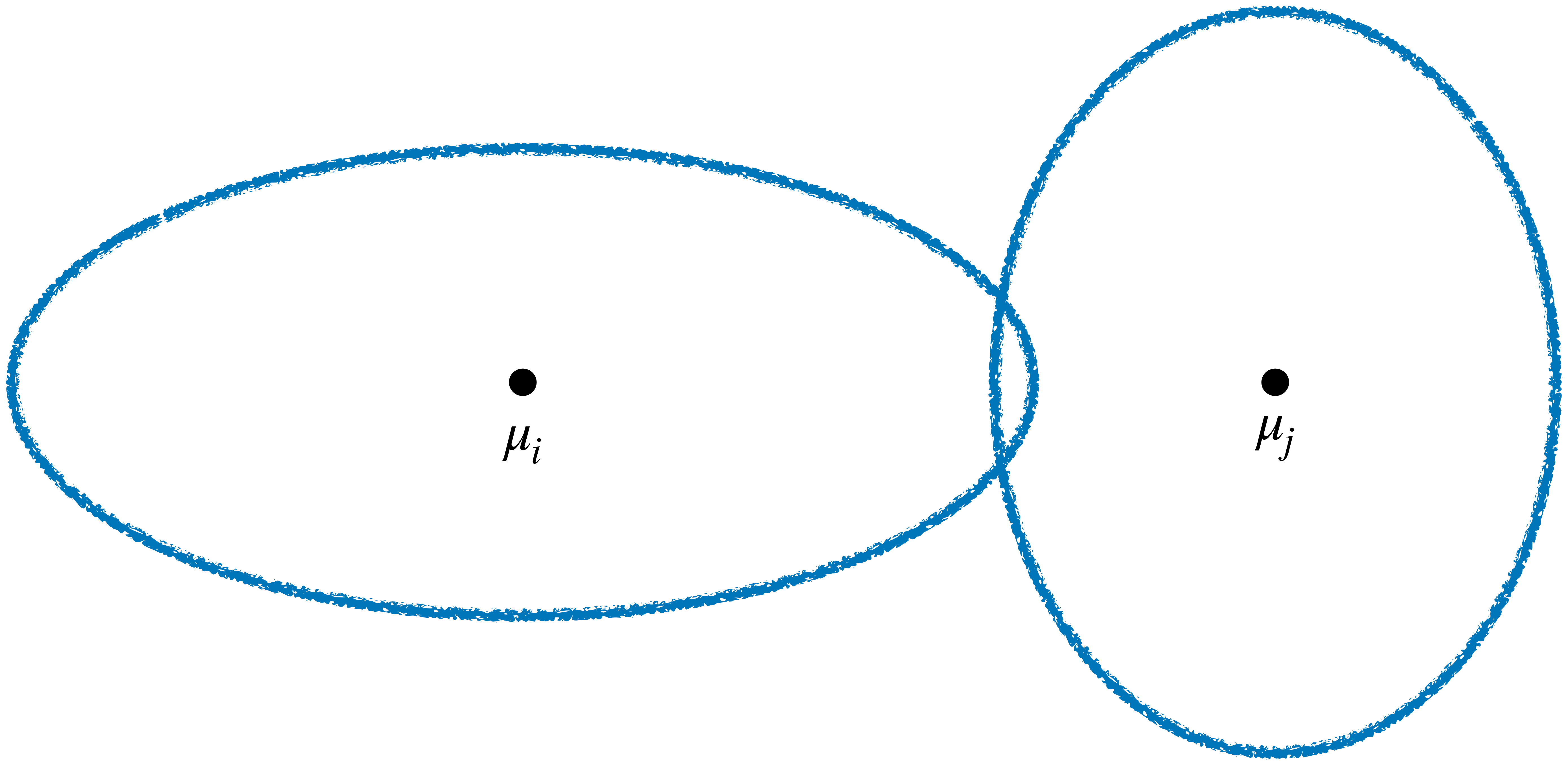


**Observation:** Suffices to find the component means, and cluster by nearest representative

# Algorithm — Uniform Mixtures

**arXiv:2312.11769**

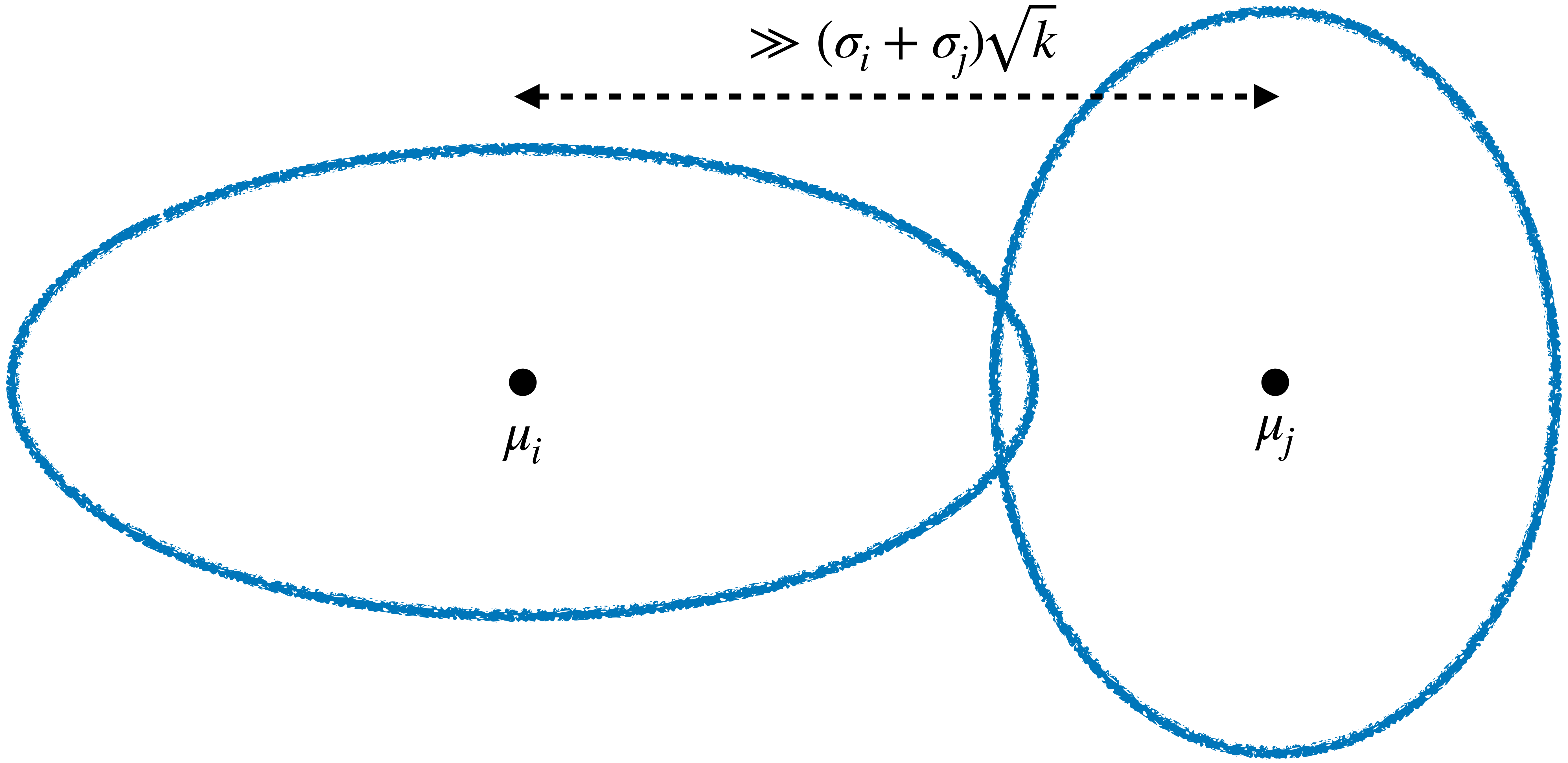
**Observation:** Suffices to find the component means, and cluster by nearest representative



# Algorithm — Uniform Mixtures

arXiv:2312.11769

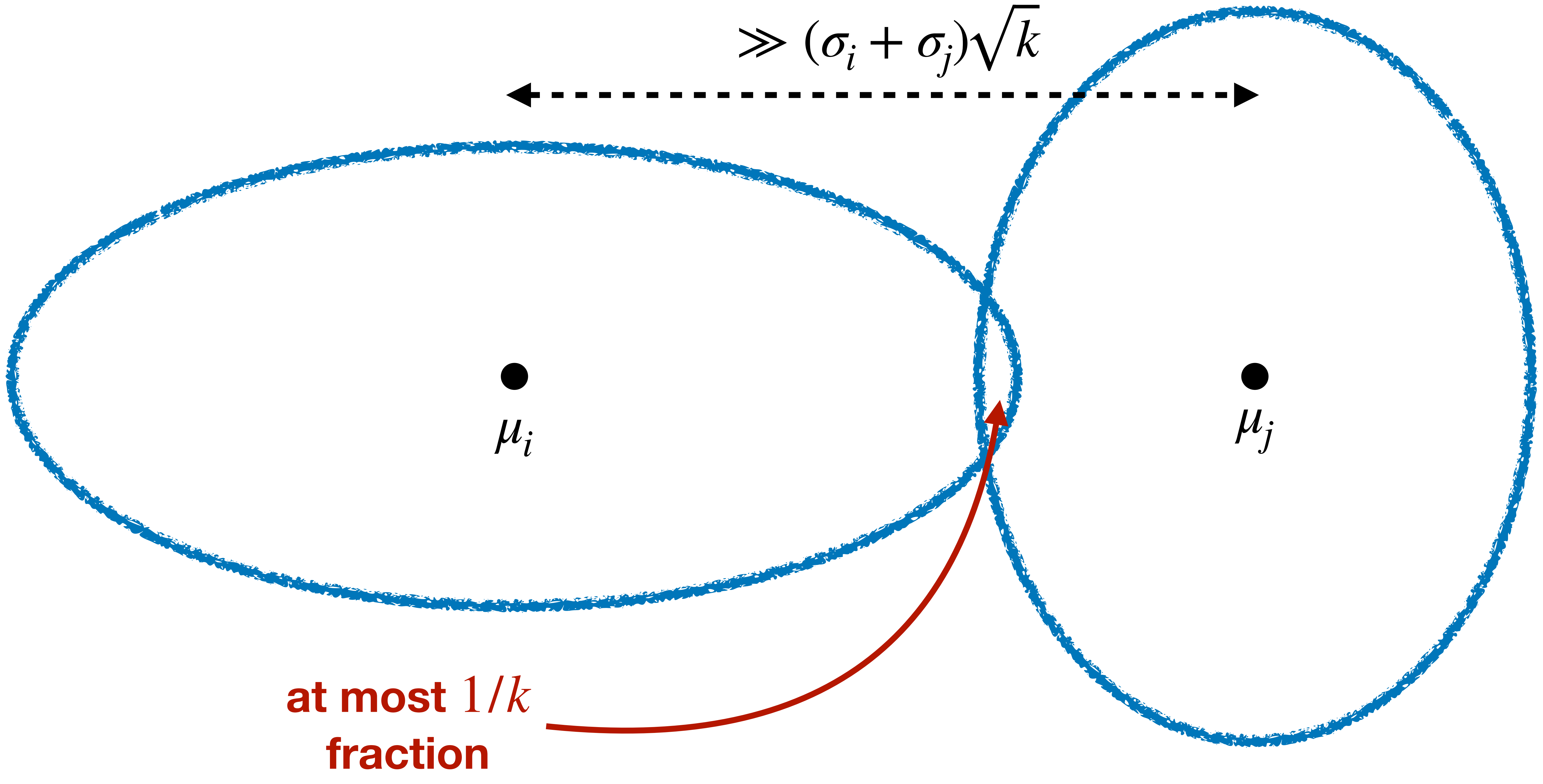
**Observation:** Suffices to find the component means, and cluster by nearest representative



# Algorithm — Uniform Mixtures

arXiv:2312.11769

**Observation:** Suffices to find the component means, and cluster by nearest representative



# Algorithm Outline

arXiv:2312.11769

**Algorithm:**

# Algorithm Outline



**Algorithm:**

- Input: $\tilde{O}((d + \log 1/\delta)\, k^2)$ samples, parameter $k$

# Algorithm Outline

arXiv:2312.11769

**Algorithm:**

- Input: $\tilde{O}((d + \log 1/\delta)\, k^2)$ samples, parameter $k$

**1.** Generate many (but poly-sized many) candidate means

# Algorithm Outline

arXiv:2312.11769

**Algorithm:**

- Input: $\tilde{O}((d + \log 1/\delta)\, k^2)$ samples, parameter $k$

**1.** Generate many (but poly-sized many) candidate means

**i.** Using list-decodable mean estimation

# Algorithm Outline

arXiv:2312.11769

**Algorithm:**

- Input: $\tilde{O}((d + \log 1/\delta)\, k^2)$ samples, parameter $k$

**1.** Generate many (but poly-sized many) candidate means

**i.** Using list-decodable mean estimation

**2.** Pruning to get exactly 1 close-enough candidate mean per cluster

# Algorithm Outline

arXiv:2312.11769

**Algorithm:**

- Input: $\tilde{O}((d + \log 1/\delta)\, k^2)$ samples, parameter $k$

**1.** Generate many (but poly-sized many) candidate means

**i.** Using list-decodable mean estimation

**2.** Pruning to get exactly 1 close-enough candidate mean per cluster

**i.** Ensure every candidate mean is close to a cluster mean

# Algorithm Outline

arXiv:2312.11769

**Algorithm:**

- Input: $\tilde{O}((d + \log 1/\delta)\, k^2)$ samples, parameter $k$

**1.** Generate many (but poly-sized many) candidate means

**i.** Using list-decodable mean estimation

**2.** Pruning to get exactly 1 close-enough candidate mean per cluster

**i.** Ensure every candidate mean is close to a cluster mean

**ii.** Prune if too many means per cluster

# List-Decodable Mean Estimation

arXiv:2312.11769

# List-Decodable Mean Estimation

arXiv:2312.11769

**Problem:** Given $\alpha n$ samples from a distribution $P$ with covariance $\preceq \sigma^2 I$,

mixed with arbitrary $(1-\alpha)n$ outliers, estimate the mean of $P$?

**What can we do when $\alpha < 1/2$?**

# List-Decodable Mean Estimation

arXiv:2312.11769

**Problem:** Given $\alpha n$ samples from a distribution $P$ with covariance $\preceq \sigma^2 I$,

mixed with arbitrary $(1-\alpha)n$ outliers, estimate the mean of $P$?

**What can we do when $\alpha < 1/2$?**

**Fact [DKKLT22]:** Near-linear time algorithm, outputs a ***list*** of $O(1/\alpha)$ vectors

**One** of them is $O(\sigma/\sqrt{\alpha})$-close to the true mean of $P$

# List-Decodable Mean Estimation

arXiv:2312.11769

**Fact [DKKLT22]:** Near-linear time algorithm, outputs a ***list*** of $O(1/\alpha)$ vectors

**One** of them is $O(\sigma/\sqrt{\alpha})$-close to the true mean of $P$

# List-Decodable Mean Estimation

arXiv:2312.11769

**Fact [DKKLT22]:** Near-linear time algorithm, outputs a ***list*** of $O(1/\alpha)$ vectors

**One** of them is $O(\sigma/\sqrt{\alpha})$-close to the true mean of $P$

**Use** $\alpha \approx 1/k$

# List-Decodable Mean Estimation

arXiv:2312.11769

**Fact [DKKLT22]:** Near-linear time algorithm, outputs a ***list*** of $O(1/\alpha)$ vectors

**One** of them is $O(\sigma/\sqrt{\alpha})$-close to the true mean of $P$

**Use** $\alpha \approx 1/k$

**Caveat:** DKKLT22 requires knowing $\sigma$ to constant factor.

# List-Decodable Mean Estimation

arXiv:2312.11769

**Fact [DKKLT22]:** Near-linear time algorithm, outputs a ***list*** of $O(1/\alpha)$ vectors

**One** of them is $O(\sigma/\sqrt{\alpha})$-close to the true mean of $P$

**Use** $\alpha \approx 1/k$

**Caveat:** DKKLT22 requires knowing $\sigma$ to constant factor.

**Solution:** First generate a $\mathrm{poly}(n)$-sized list of candidate $\sigma_i$,

then run DKKLT22 using all candidate standard deviations

# Algorithm Outline

arXiv:2312.11769

**Algorithm:**

- Input: $\tilde{O}((d + \log 1/\delta)\, k^2)$ samples, parameter $k$

**1.** Generate many (but poly-sized many) candidate means + s.d.

**i.** Using list-decodable mean estimation

**2.** Pruning to get exactly 1 close-enough candidate mean per cluster

**i.** Ensure every candidate mean is close to a cluster mean

**ii.** Prune if too many means per cluster

# Pruning — Main Step

arXiv:2312.11769

Can be $O(\hat{s}\sqrt{k})$ from true cluster mean

**Ingredient:** Check if candidate mean $\hat{\mu}$ corresponds to cluster of $\approx n/k$ samples w/ standard deviation $\hat{s}$

# Pruning — Main Step

arXiv:2312.11769

Can be $O(\hat{s}\sqrt{k})$ from true cluster mean

**Ingredient:** Check if candidate mean $\hat{\mu}$ corresponds to cluster of $\approx n/k$ samples w/ standard deviation $\hat{s}$

Find: $w_x \in [0,1]$ for all $x$ in sample set

such that
$$\left\| \sum_x w_x \left(x - \sum_y w_y y\right)\left(x - \sum_y w_y y\right)^\top \right\|_{\mathrm{op}} \le O(\hat{s}^2) \sum_x w_x$$

$$\sum_x w_x \ge 0.97 n/k \qquad \left\| \sum_x w_x x - \hat{\mu} \right\|_2 \le O(\hat{s}\sqrt{k})$$

# Pruning — Main Step

arXiv:2312.11769

Can be $O(\hat{s}\sqrt{k})$ from true cluster mean

**Ingredient:** Check if candidate mean $\hat{\mu}$ corresponds to cluster of $\approx n/k$ samples w/ standard deviation $\hat{s}$

Find: $w_x \in [0,1]$ for all $x$ in sample set

such that $$\left\| \sum_x w_x \left(x - \sum_y w_y y\right)\left(x - \sum_y w_y y\right)^\top \right\|_{\mathrm{op}} \leq O(\hat{s}^2) \sum_x w_x$$

Non-convex!

$$\sum_x w_x \geq 0.97 n/k \qquad \left\| \sum_x w_x x - \hat{\mu} \right\|_2 \leq O(\hat{s}\sqrt{k})$$

# Pruning — Main Step

arXiv:2312.11769

**Ingredient:** Check if candidate mean $\hat{\mu}$ corresponds to cluster of $\approx n/k$ samples w/ standard deviation $\hat{s}$

Find: $w_x \in [0,1]$ for all $x$ in sample set

such that
$$\left\| \sum_x w_x (x-\hat{\mu})(x-\hat{\mu})^\top \right\|_{(k)} \le O(\hat{s}^2 k) \sum_x w_x$$

$$\sum_x w_x \ge 0.97n/k$$

Ky-Fan norm = sum of top-$k$ singular/eigenvalues

# Algorithm Outline

arXiv:2312.11769

**Algorithm:**

- Input: $\tilde{O}((d + \log 1/\delta)\, k^2)$ samples, parameter $k$

**1.** Generate many (but poly-sized many) candidate means + s.d. ✓

**i.** Using list-decodable mean estimation

**2.** Pruning to get exactly 1 close-enough candidate mean per cluster

**i.** Ensure every candidate mean is close to a cluster mean ✓

**ii.** Prune if too many means per cluster

# Pruning — By Mass



**Issue:** A cluster can correspond to many remaining candidate means

# Pruning — By Mass



**Issue:** A cluster can correspond to many remaining candidate means

**Observation:** Multiple candidate means will ***split*** a cluster, at least one with small size ( $\ll 1/k$ fraction)

# Pruning — By Mass

arXiv:2312.11769

**Issue:** A cluster can correspond to many remaining candidate means

**Observation:** Multiple candidate means will ***split*** a cluster,
at least one with small size ( $\ll 1/k$ fraction)

**Solution:** Repeatedly cluster with nearest representative,
and prune candidate means with cluster size $\leq 0.96n/k$

# Algorithm Outline

arXiv:2312.11769

**Algorithm:**

- Input: $\tilde{O}((d + \log 1/\delta)\, k^2)$ samples, parameter $k$

**1.** Generate many (but poly-sized many) candidate means + s.d. ✓

**i.** Using list-decodable mean estimation

**2.** Pruning to get exactly 1 close-enough candidate mean per cluster

**i.** Ensure every candidate mean is close to a cluster mean ✓

**ii.** Prune if too many means per cluster ✓

# Robust Clustering Mixture Distributions

arXiv:2312.11769

$\epsilon \leq \alpha/100$

$w_i \geq \alpha$

**Setup:**

- $\epsilon$-contaminated samples from $k$-mixture $D = \sum_{i=1}^{k} w_i P_i$
- $P_i$ has mean $\mu_i$ and covariance $\Sigma_i \preceq \sigma_i^2 I$, both unknown
- $\mu_i, \mu_j$ "well separated" $\leftarrow$ $\|\mu_i - \mu_j\| \gg (\sigma_i + \sigma_j)/\sqrt{\alpha}$

**Goal:** Identify 95% of the samples correctly for **every** cluster

# Robust Clustering Mixture Distributions

arXiv:2312.11769

$\epsilon \leq \alpha/100$ $\qquad\qquad$ $w_i \geq \alpha$

**Setup:**

- $\epsilon$-contaminated samples from $k$-mixture $D = \sum_{i=1}^{k} w_i P_i$
- $P_i$ has mean $\mu_i$ and covariance $\Sigma_i \preceq \sigma_i^2 I$, both unknown
- $\mu_i, \mu_j$ "well separated" ← $\|\mu_i - \mu_j\| \gg (\sigma_i + \sigma_j)/\sqrt{\alpha}$

**Goal:** Identify 95% of the samples correctly for every cluster

**Impossible**

# Non-identifiability

# Non-identifiability

arXiv:2312.11769

**Even if we know**:

- $k = 3$
- Min weight $\alpha = 1/4$

# Non-identifiability

arXiv:2312.11769

**Even if we know:**

- $k = 3$
- Min weight $\alpha = 1/4$

**Even with infinite uncorrupted samples**

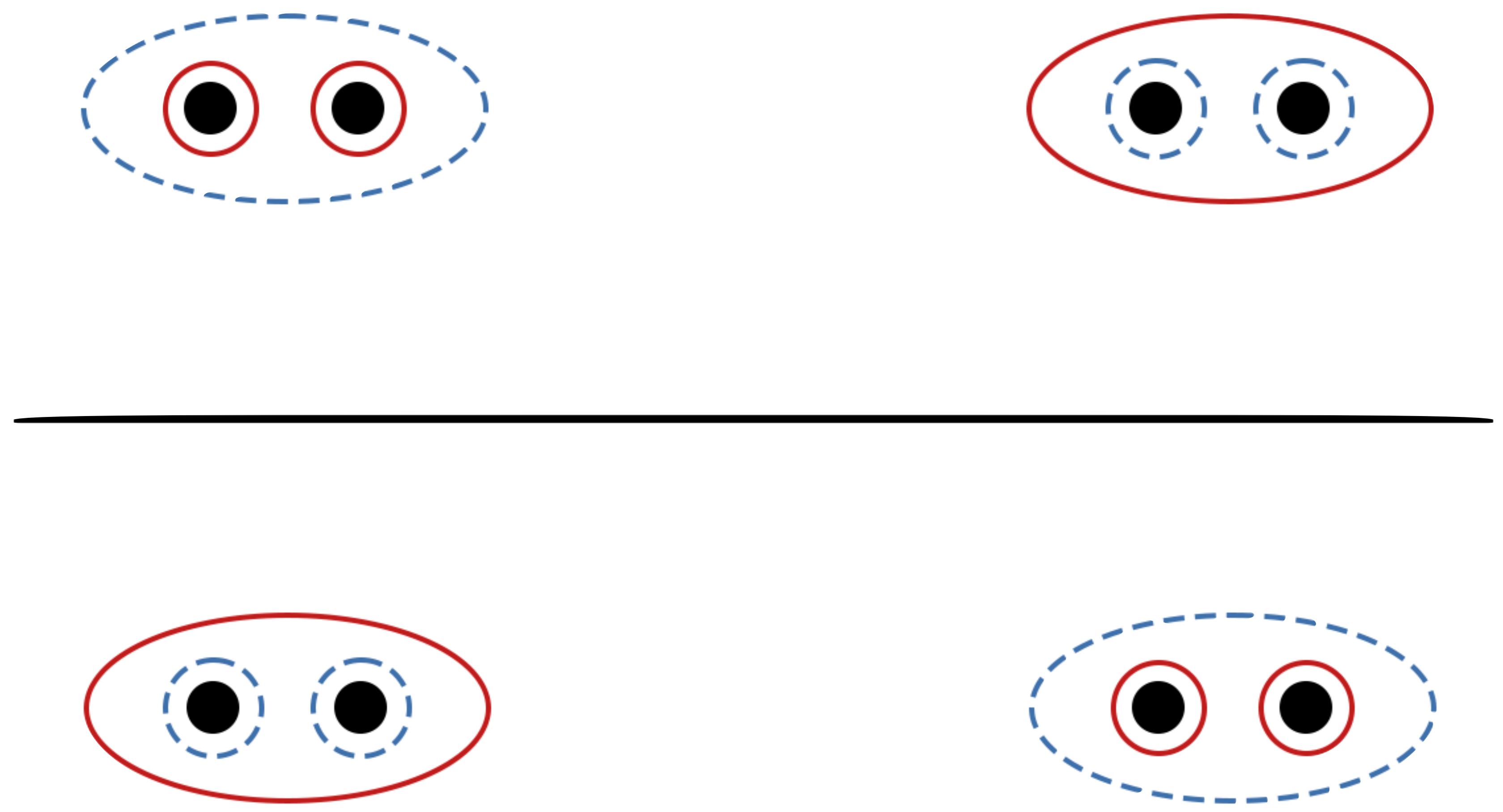
# Non-identifiability

arXiv:2312.11769

**Even if we know:**

- $k = 3$
- Min weight $\alpha = 1/4$

**Even with infinite uncorrupted samples**



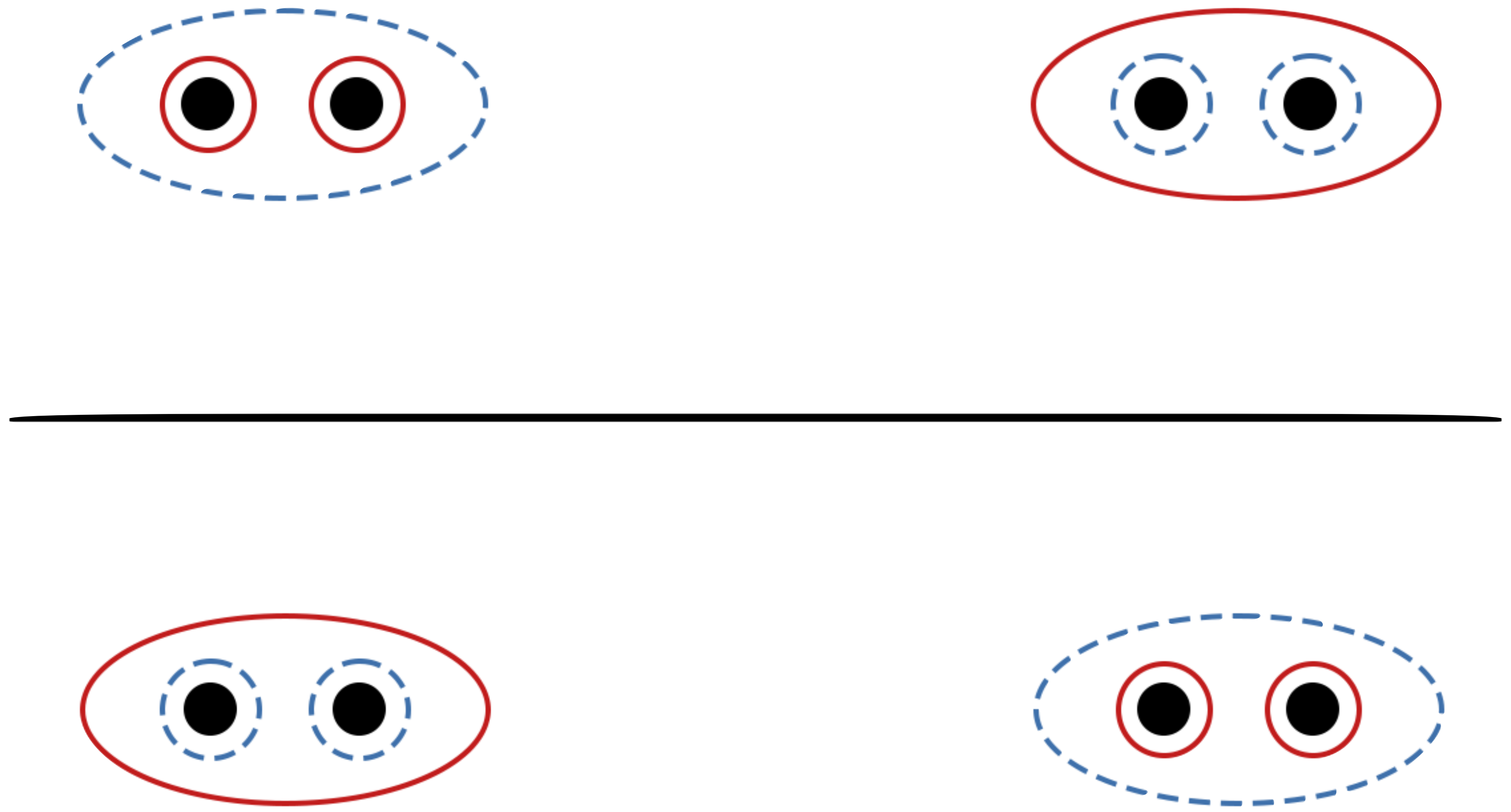
# Non-identifiability

arXiv:2312.11769

**Even if we know:**

- $k = 3$
- Min weight $\alpha = 1/4$

**Even with infinite uncorrupted samples**



**Now what?**

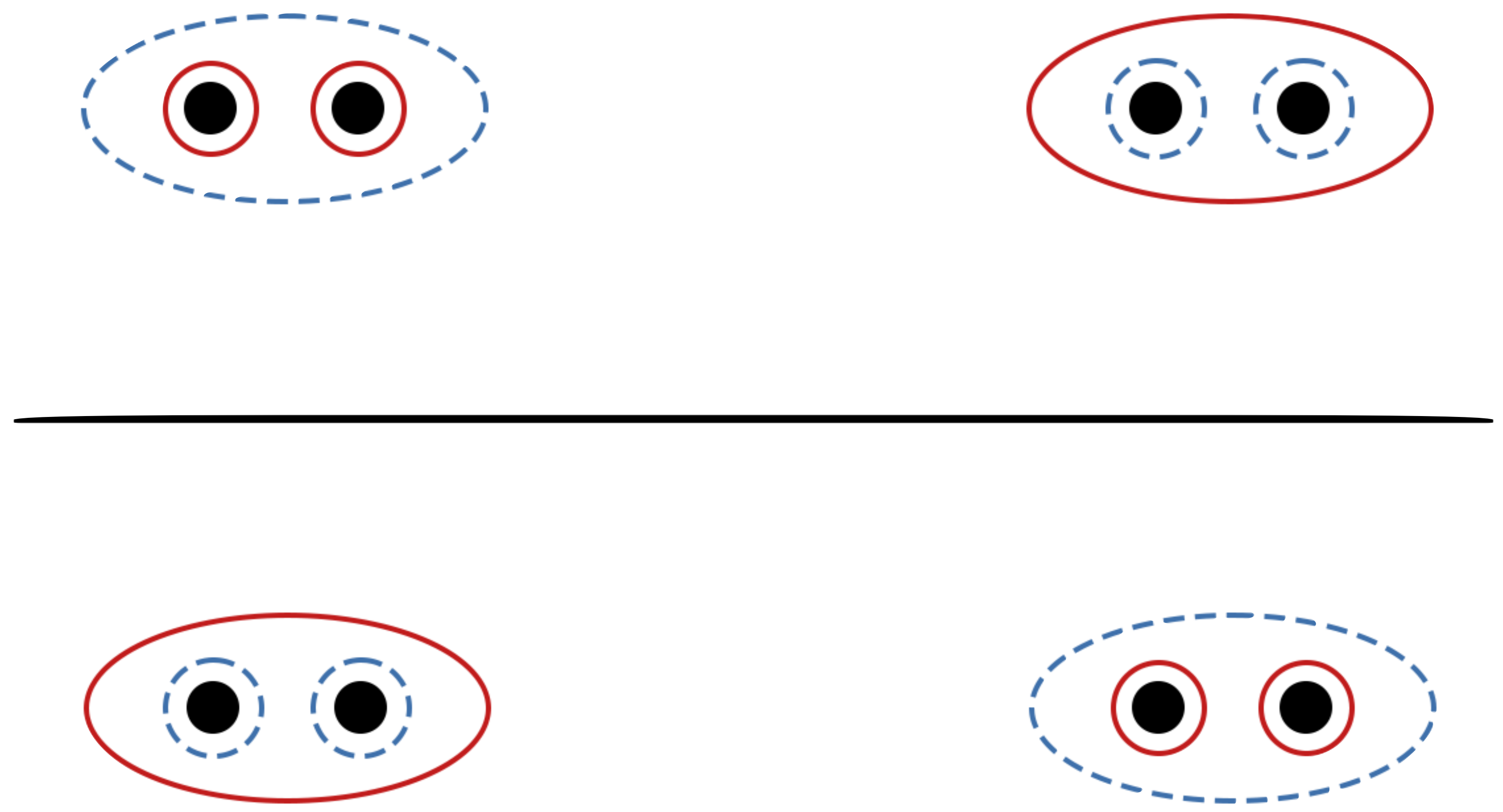
# Non-identifiability

arXiv:2312.11769

**Even if we know:**

- $k = 3$
- Min weight $\alpha = 1/4$

**Even with infinite uncorrupted samples**



**Question:** What information ***can*** we compute about the clustering?

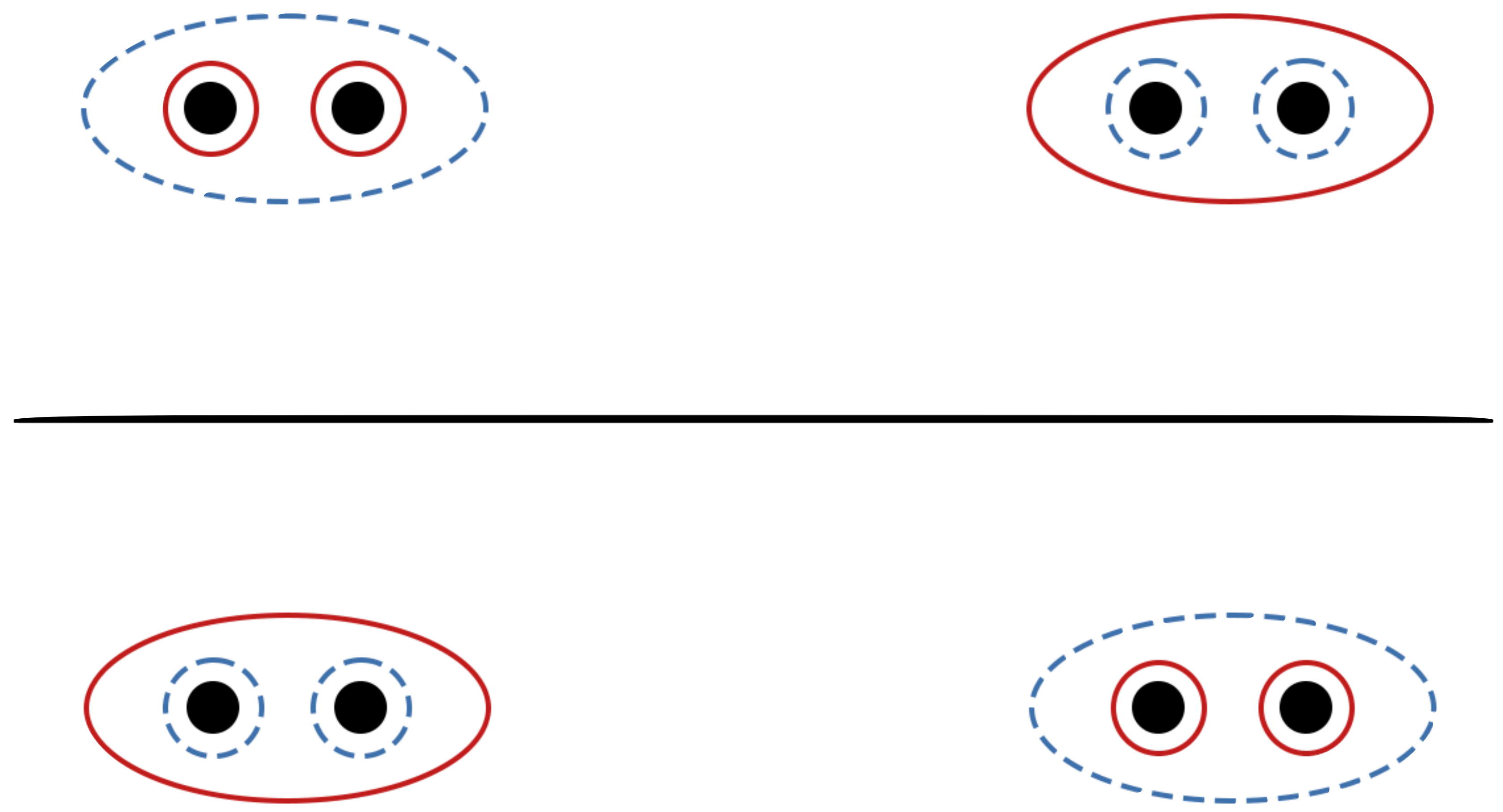
# Non-identifiability

arXiv:2312.11769

**Even if we know:**

- $k = 3$
- Min weight $\alpha = 1/4$

**Even with infinite uncorrupted samples**



**Question:** What information ***can*** we compute about the clustering?

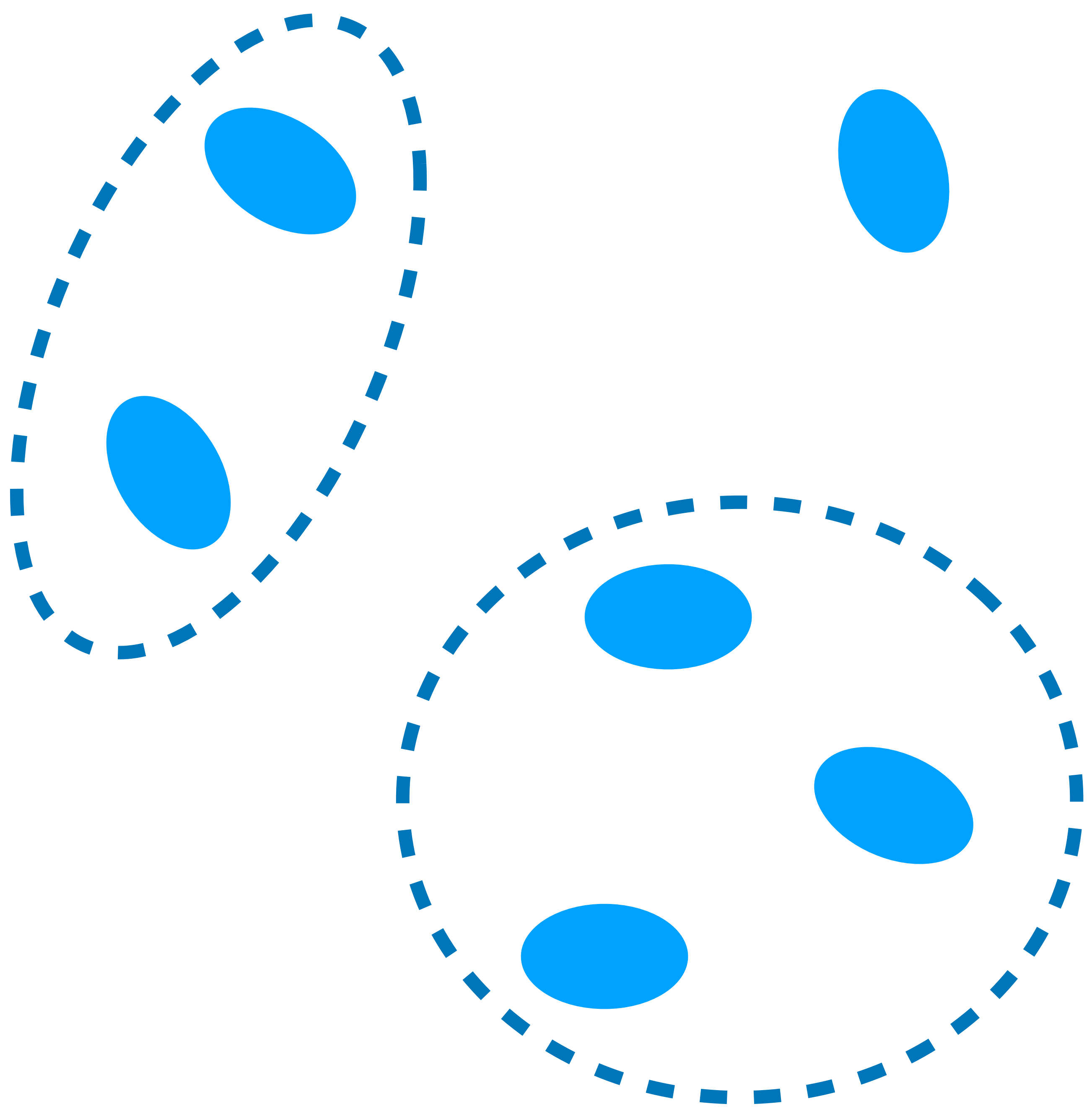
**Maybe:** Compute all sub-clusterings, except for the grouping

# Clustering Refinement

arXiv:2312.11769

**Definition:** Given true cluster samples $S_1, \ldots, S_k$, totalling $n$ samples, the disjoint subsets $B_1, \ldots, B_m$ form an ***accurate refinement*** if:

- $|B_j| \geq 0.95\alpha n$
- $\|\mu_{B_j} - \mu_{B_{j'}}\| \gg (\sigma_{B_j} + \sigma_{B_{j'}})/\sqrt{\alpha}$
- They can be grouped into $k$ sample sets $S'_1, \ldots, S'_k$ such that
    - $S_i$ and $S'_i$ have 92% overlap



# Theorem — Arbitrary Mixtures

arXiv:2312.11769

**Failure probability** **Corrupted,** $\epsilon \leq \alpha/100$ $w_i \geq \alpha$

**Theorem:** Given $\tilde{O}((d + \log 1/\delta)/\alpha^2)$ samples from $D = \sum_i w_i P_i$

where $P_i$ has mean $\mu_i$ and covariance $\Sigma_i \preceq \sigma_i^2 I$ (all unknown)

and $\|\mu_i - \mu_j\| \gg (\sigma_i + \sigma_j)/\sqrt{\alpha}$

***Algorithm*** returns sets $B_1, \ldots, B_m$ that is an ***accurate refinement***

of true clustering $S_1, \ldots, S_k$

# Theorem — Arbitrary Mixtures

arXiv:2312.11769

Failure probability

Corrupted, $\epsilon \leq \alpha/100$

$w_i \geq \alpha$

**Theorem:** Given $\tilde{O}((d + \log 1/\delta)/\alpha^2)$ samples from $D = \sum_i w_i P_i$

where $P_i$ has mean $\mu_i$ and covariance $\Sigma_i \preceq \sigma_i^2 I$ (all unknown)

and $\|\mu_i - \mu_j\| \gg (\sigma_i + \sigma_j)/\sqrt{\alpha}$

***Algorithm*** returns sets $B_1, \ldots, B_m$ that is an ***accurate refinement***

of true clustering $S_1, \ldots, S_k$

**Remarks**:

# Theorem — Arbitrary Mixtures

**arXiv:2312.11769**

**Failure probability** **Corrupted,** $\epsilon \leq \alpha/100$ $w_i \geq \alpha$

**Theorem:** Given $\tilde{O}((d + \log 1/\delta)/\alpha^2)$ samples from $D = \sum_i w_i P_i$

where $P_i$ has mean $\mu_i$ and covariance $\Sigma_i \preceq \sigma_i^2 I$ (all unknown)

and $\|\mu_i - \mu_j\| \gg (\sigma_i + \sigma_j)/\sqrt{\alpha}$

***Algorithm*** returns sets $B_1, \ldots, B_m$ that is an ***accurate refinement***

of true clustering $S_1, \ldots, S_k$

**Remarks**:

- One ***single*** algorithm for both theorems

# Theorem — Arbitrary Mixtures

**arXiv:2312.11769**

**Failure probability** **Corrupted, $\epsilon \leq \alpha/100$** $w_i \geq \alpha$

**Theorem:** Given $\tilde{O}((d + \log 1/\delta)/\alpha^2)$ samples from $D = \sum_i w_i P_i$

where $P_i$ has mean $\mu_i$ and covariance $\Sigma_i \preceq \sigma_i^2 I$ (all unknown)

and $\|\mu_i - \mu_j\| \gg (\sigma_i + \sigma_j)/\sqrt{\alpha}$

***Algorithm*** returns sets $B_1, \ldots, B_m$ that is an ***accurate refinement***

of true clustering $S_1, \ldots, S_k$

**Remarks**:

**Previous alg (replace $k$ with $1/\alpha$) + distance-based pruning**

- One ***single*** algorithm for both theorems

# Theorem — Arbitrary Mixtures

**arXiv:2312.11769**

**Failure probability**

**Corrupted, $\epsilon \leq \alpha/100$**

$w_i \geq \alpha$

**Theorem:** Given $\tilde{O}((d + \log 1/\delta)/\alpha^2)$ samples from $D = \sum_i w_i P_i$

where $P_i$ has mean $\mu_i$ and covariance $\Sigma_i \preceq \sigma_i^2 I$ (all unknown)

and $\|\mu_i - \mu_j\| \gg (\sigma_i + \sigma_j)/\sqrt{\alpha}$

***Algorithm*** returns sets $B_1, \ldots, B_m$ that is an ***accurate refinement***

of true clustering $S_1, \ldots, S_k$

**Remarks**:

**Previous alg (replace $k$ with $1/\alpha$) + distance-based pruning**

- One ***single*** algorithm for both theorems
- **Corollary:** existence of a **common refinement** for all possible clusterings

# Clustering Arbitrary Mixtures

arXiv:2312.11769

# Clustering Arbitrary Mixtures

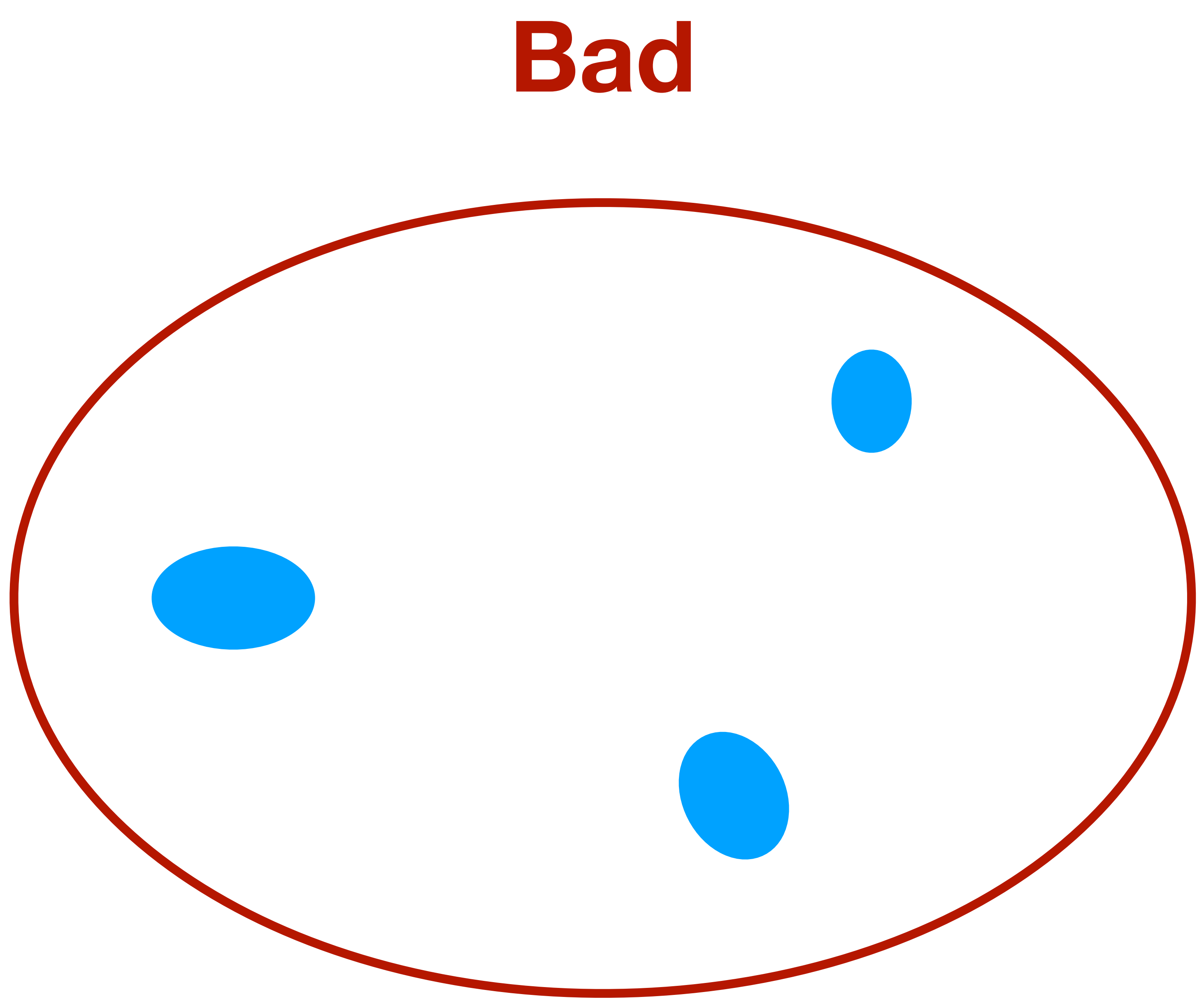
arXiv:2312.11769

**Question:** What is a **sufficient** assumption to compute a clustering not just a refinement?

# Clustering Arbitrary Mixtures

arXiv:2312.11769

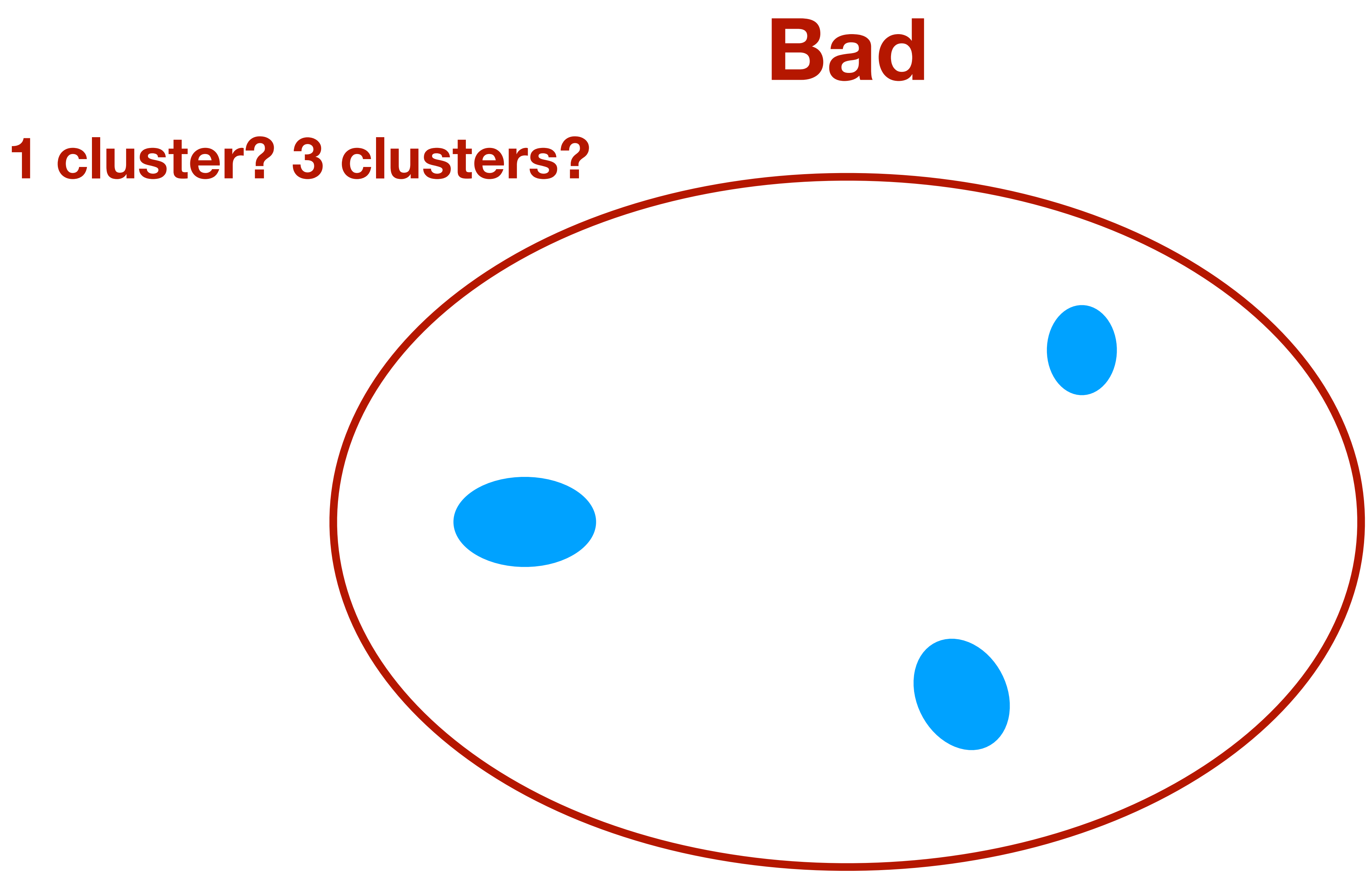
**Question:** What is a **sufficient** assumption to compute a clustering not just a refinement?



# Clustering Arbitrary Mixtures

arXiv:2312.11769

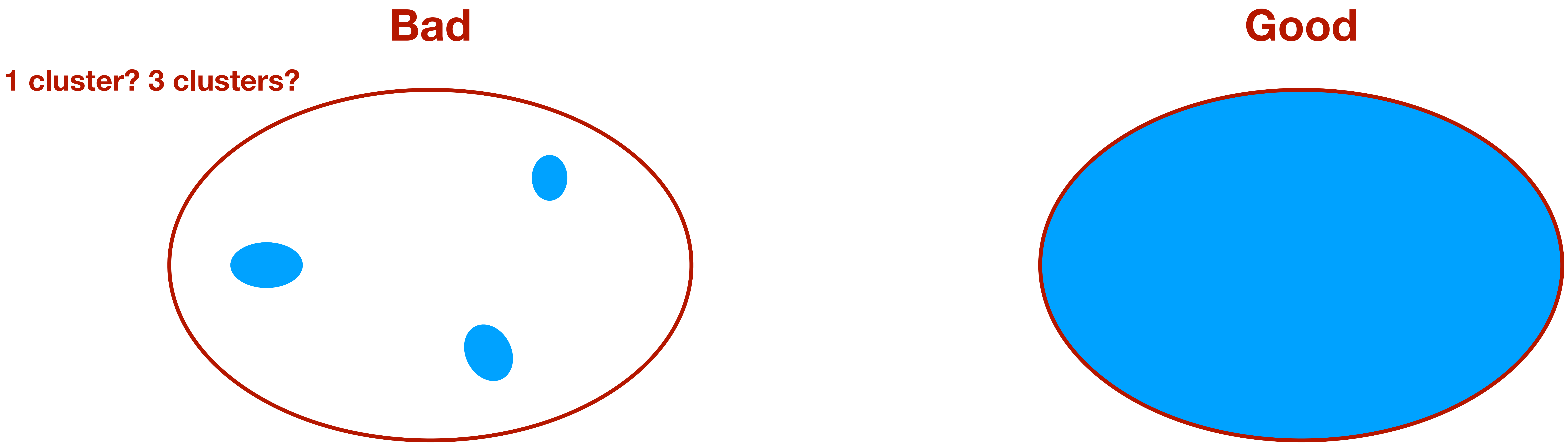
**Question:** What is a **sufficient** assumption to compute a clustering not just a refinement?



# Clustering Arbitrary Mixtures

arXiv:2312.11769

**Question:** What is a **sufficient** assumption to compute a clustering not just a refinement?



# No Large Sub-Cluster Condition

arXiv:2312.11769

**Definition:** The sample sets $S_1, \ldots, S_k$ of total size $n$ have "no large sub-clusters" if

For every $S_i$ and every subset $S' \subseteq S_i$ of size $\geq 0.8\alpha n$ ← **Every large subset**

We have $\sigma_{S'} \geq 0.1\sigma_{S_i}$ ← **Should not look like its own cluster**

# No Large Sub-Cluster Condition

arXiv:2312.11769

**Definition:** The sample sets $S_1, \ldots, S_k$ of total size $n$ have "no large sub-clusters" if

For every $S_i$ and every subset $S' \subseteq S_i$ of size $\geq 0.8\alpha n$ ← **Every large subset**

We have $\sigma_{S'} \geq 0.1\sigma_{S_i}$ ← **Should not look like its own cluster**

**Theorem:** If the (uncorrupted) input samples have no large sub-clusters,

then ***Algorithm*** returns a clustering with $k$ sets instead of a refinement.

# No Large Sub-Cluster Condition

arXiv:2312.11769

**Definition:** The sample sets $S_1, \ldots, S_k$ of total size $n$ have "no large sub-clusters" if

For every $S_i$ and every subset $S' \subseteq S_i$ of size $\geq 0.8\alpha n$ ← **Every large subset**

We have $\sigma_{S'} \geq 0.1\sigma_{S_i}$ ← **Should not look like its own cluster**

**Proposition:** For well-conditioned+high-d log-concave distributions, drawing $\tilde{O}(d/\alpha^2)$ samples ensures no large sub-clusters, due to thin-shell behavior.

# No Large Sub-Cluster Condition

arXiv:2312.11769

**Definition:** The sample sets $S_1, \ldots, S_k$ of total size $n$ have "no large sub-clusters" if

For every $S_i$ and every subset $S' \subseteq S_i$ of size $\geq 0.8\alpha n$ ← **Every large subset**

We have $\sigma_{S'} \geq 0.1\sigma_{S_i}$ ← **Should not look like its own cluster**

≈ **isotropic covariance**

**Proposition:** For well-conditioned+high-d log-concave distributions, drawing $\tilde{O}(d/\alpha^2)$ samples ensures no large sub-clusters, due to thin-shell behavior.

# No Large Sub-Cluster Condition

arXiv:2312.11769

**Definition:** The sample sets $S_1, \ldots, S_k$ of total size $n$ have "no large sub-clusters" if

For every $S_i$ and every subset $S' \subseteq S_i$ of size $\geq 0.8\alpha n$ ← **Every large subset**

We have $\sigma_{S'} \geq 0.1\sigma_{S_i}$ ← **Should not look like its own cluster**

$\approx$ **isotropic covariance** $d \geq \text{polylog}(1/\alpha)$

**Proposition:** For well-conditioned+high-d log-concave distributions, drawing $\tilde{O}(d/\alpha^2)$ samples ensures no large sub-clusters, due to thin-shell behavior.

# Summary

arXiv:2312.11769

# Summary

arXiv:2312.11769

$w_i \geq \alpha$

**Problem:** Cluster samples from $\sum_i w_i P_i$ under fine-grained separation $\|\mu_i - \mu_j\| \gg (\sigma_i + \sigma_j)/\sqrt{\alpha}$

**Mean $\mu_i$, Covariance $\Sigma_i \preceq \sigma_i^2 I$**

# Summary

arXiv:2312.11769

$w_i \geq \alpha$

**Problem:** Cluster samples from $\sum_i w_i P_i$ under fine-grained separation $\|\mu_i - \mu_j\| \gg (\sigma_i + \sigma_j)/\sqrt{\alpha}$

**Mean $\mu_i$, Covariance $\Sigma_i \preceq \sigma_i^2 I$**

**A single poly-time algorithm such that:**

- Near-uniform mixture: recovers clustering to 95% accuracy
- Arbitrary mixtures: recovers ***accurate refinement***
- Arbitrary mixture + No Large Sub-Cluster condition: recovers clustering to 95% accuracy
- Can tolerate corruption level $\epsilon \leq \alpha/100$

# Summary

arXiv:2312.11769

$w_i \geq \alpha$

**Problem:** Cluster samples from $\sum_i w_i P_i$ under fine-grained separation $\|\mu_i - \mu_j\| \gg (\sigma_i + \sigma_j)/\sqrt{\alpha}$

**Mean $\mu_i$, Covariance $\Sigma_i \preceq \sigma_i^2 I$**

**A single poly-time algorithm such that:**

- Near-uniform mixture: recovers clustering to 95% accuracy
- Arbitrary mixtures: recovers ***accurate refinement***
- Arbitrary mixture + No Large Sub-Cluster condition: recovers clustering to 95% accuracy
- Can tolerate corruption level $\epsilon \leq \alpha/100$

**Structural**:

- All ground truth clusterings of a mixture share a common refinement

# Open Problems

arXiv:2312.11769

# Open Problems

arXiv:2312.11769

**Computational**:

- Current algorithm is poly-time but very slow

# Open Problems

arXiv:2312.11769

**Computational**:

- Current algorithm is poly-time but very slow

**How far can we push separation assumption?**:

- Even for uniform mixtures, assumes $\leq 1/k$ pairwise overlap

# Open Problems

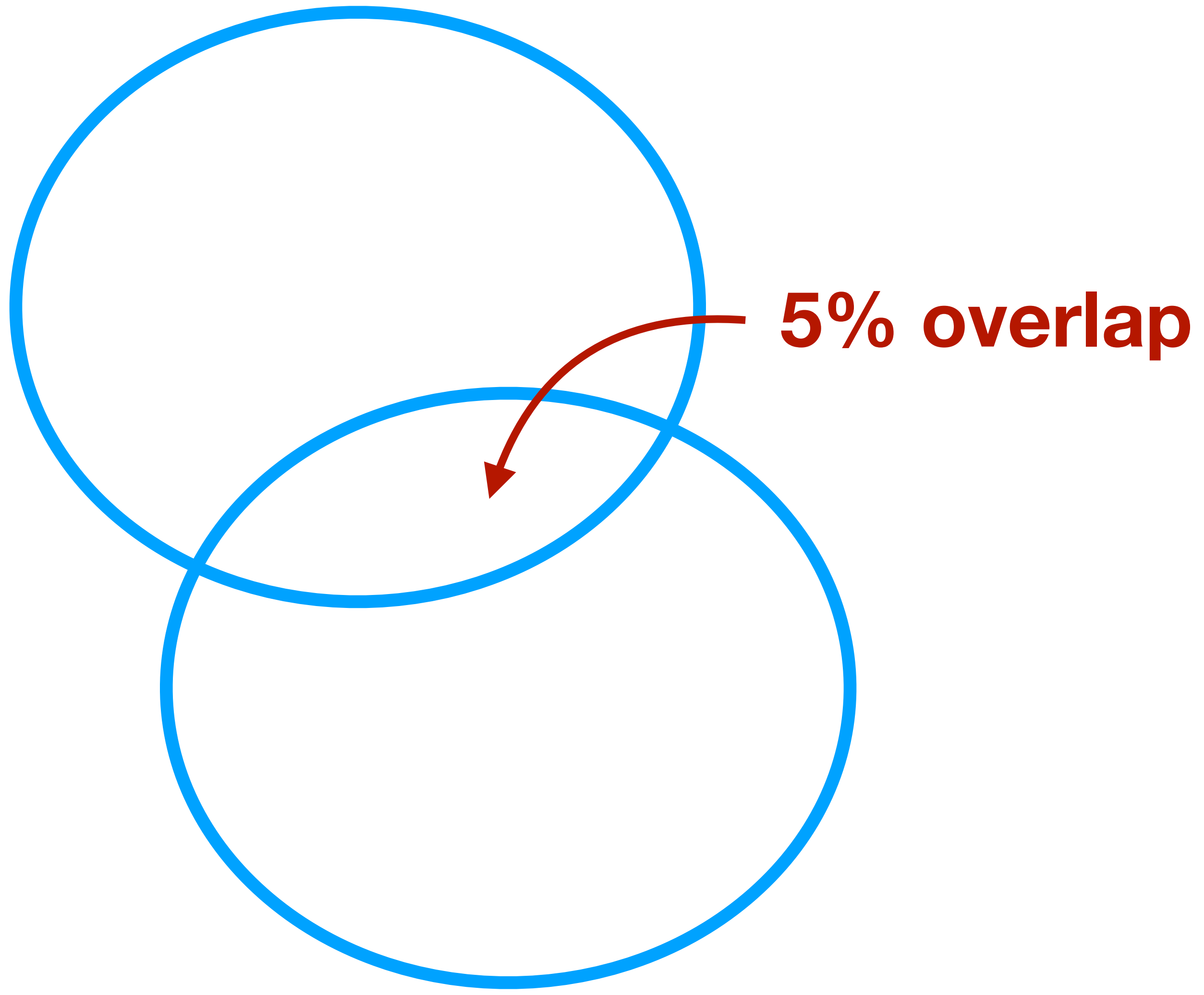
arXiv:2312.11769

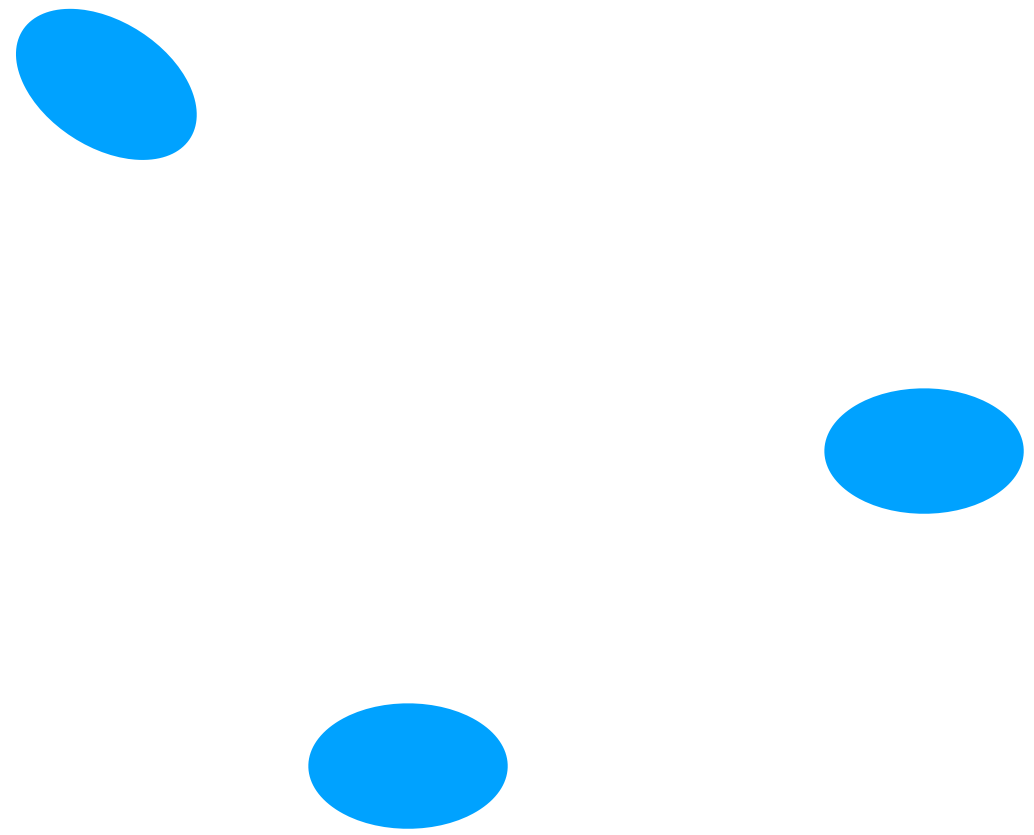
**Computational**:

- Current algorithm is poly-time but very slow

**How far can we push separation assumption?**:

- Even for uniform mixtures, assumes $\leq 1/k$ pairwise overlap



# Open Problems

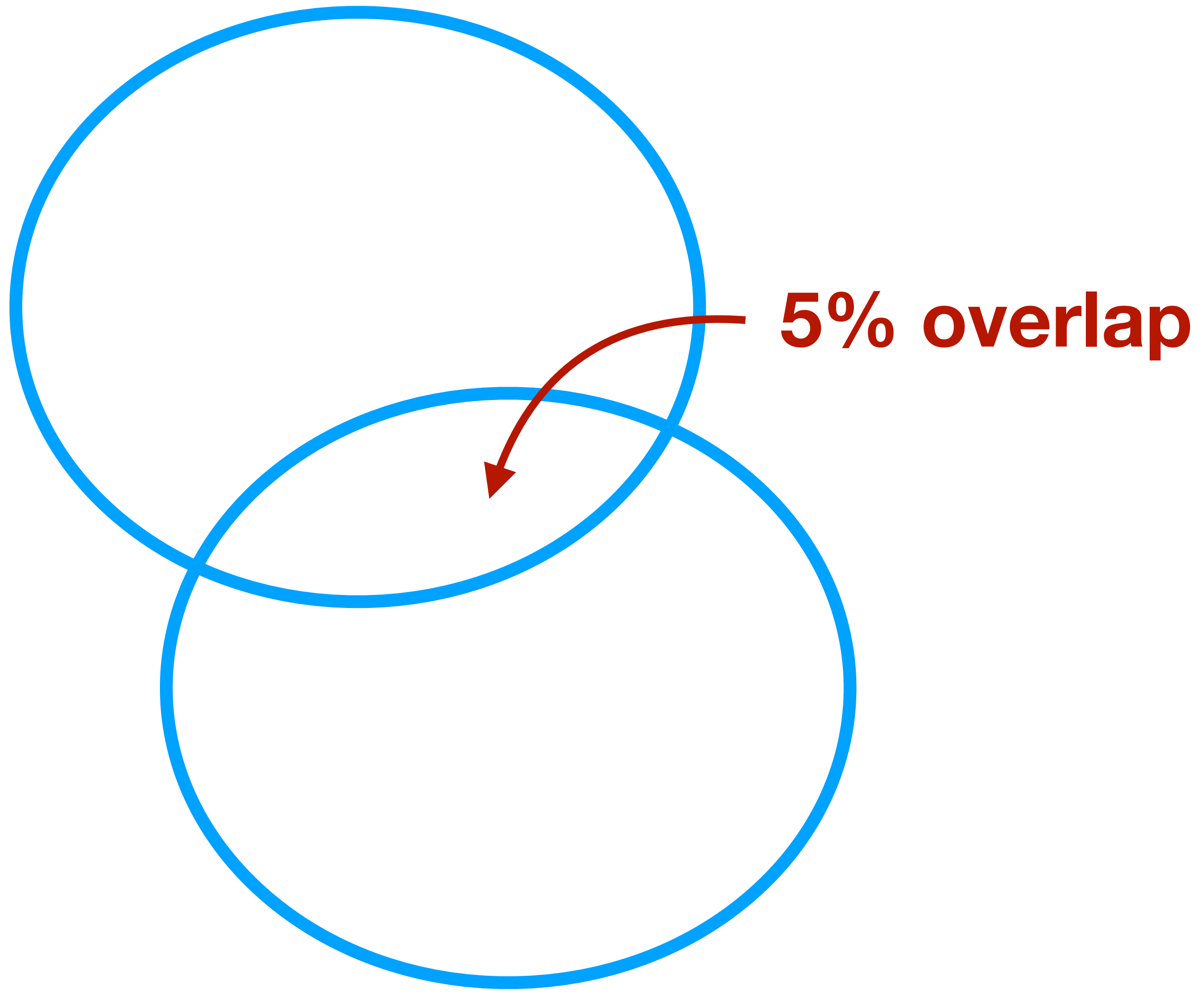
arXiv:2312.11769

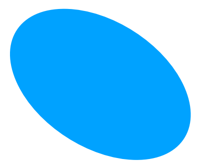
**Computational**:

- Current algorithm is poly-time but very slow

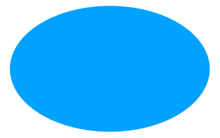
**How far can we push separation assumption?**:

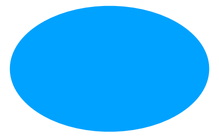
- Even for uniform mixtures, assumes $\leq 1/k$ pairwise overlap



**Goal:** Design the most ***versatile*** algorithm





# Summary

arXiv:2312.11769

$w_i \geq \alpha$

**Problem:** Cluster samples from $\sum_i w_i P_i$ under fine-grained separation $\|\mu_i - \mu_j\| \gg (\sigma_i + \sigma_j)/\sqrt{\alpha}$

**Mean $\mu_i$, Covariance $\Sigma_i \preceq \sigma_i^2 I$**

**A single poly-time algorithm such that:**

- Near-uniform mixture: recovers clustering to 95% accuracy
- Arbitrary mixtures: recovers ***accurate refinement***
- Arbitrary mixture + No Large Sub-Cluster condition: recovers clustering to 95% accuracy
- Can tolerate corruption level $\epsilon \leq \alpha/100$

**Structural**:

- All ground truth clusterings of a mixture share a common refinement