中國人民大學
RENMIN UNIVERSITY OF CHINA

Gaussian Mixture Reduction with Composite Transportation Divergence



Qiong Zhang
Renmin University of China

Archer Gong Zhang
University of Toronto

Jiahua Chen
University of British Columbia

IEEE Transactions on Information Theory (2023)
https://arxiv.org/pdf/2002.08410.pdf

Background: Gaussian mixture

- Finite Gaussian mixture density: a **convex combination** of **finitely** many **distinct** Gaussian densities

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

Background: Gaussian mixture

- Finite Gaussian mixture density: a **convex combination** of **finitely** many **distinct** Gaussian densities

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

Mixing distribution

$$G = \sum_{k=1}^{K} w_k \delta_{\theta_k}$$

Background: Gaussian mixture

- Finite Gaussian mixture density: a **convex combination** of **finitely** many **distinct** Gaussian densities

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

Mixing distribution

Component parameter

$$G = \sum_{k=1}^{K} \boxed{w_k} \delta_{\theta_k}$$

Mixing weight

Background: Gaussian mixture

- Finite Gaussian mixture density: a **convex combination** of **finitely** many **distinct** Gaussian densities

Order

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

Mixing distribution

Component parameter

$$G = \sum_{k=1}^{K} w_k \delta_{\theta_k}$$

Mixing weight

Background: Gaussian mixture

- Finite Gaussian mixture density: a **convex combination** of **finitely** many **distinct** Gaussian densities

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

Order (K); Mixing distribution (G); Component parameter (θ_k)

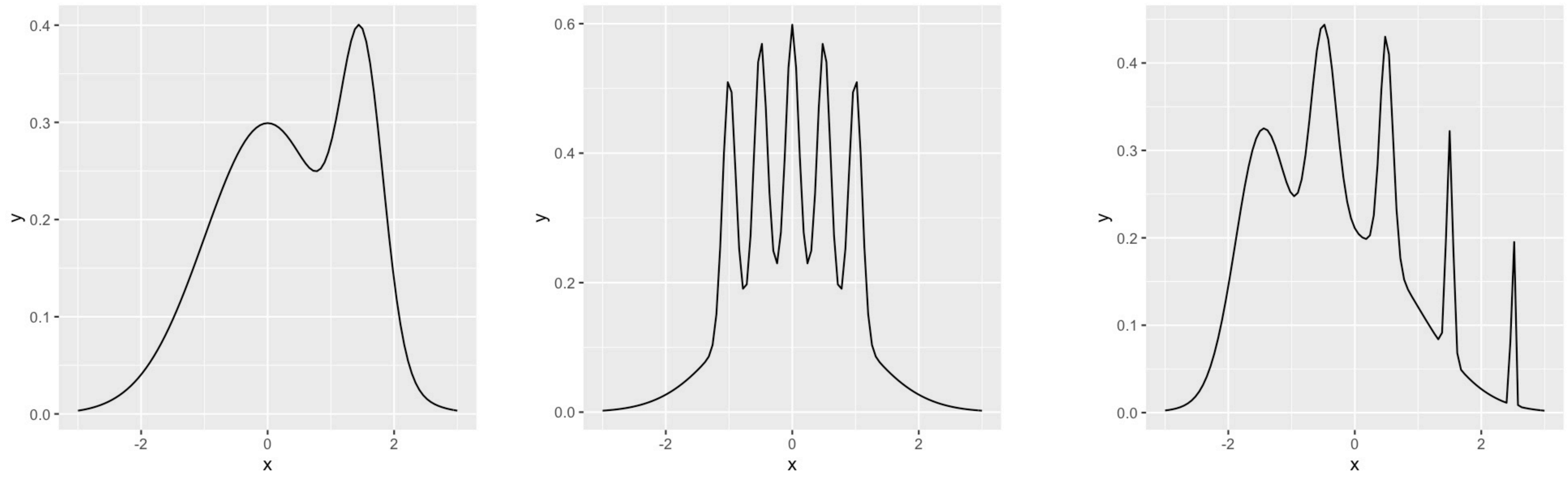
$$G = \sum_{k=1}^{K} w_k \delta_{\theta_k}$$

Mixing weight (w_k)

- **Universal approximation:** Gaussian mixture can approximate almost any smooth density functions arbitrarily well

Background: Gaussian mixture

- Finite Gaussian mixture density: a **convex combination** of **finitely** many **distinct** Gaussian densities

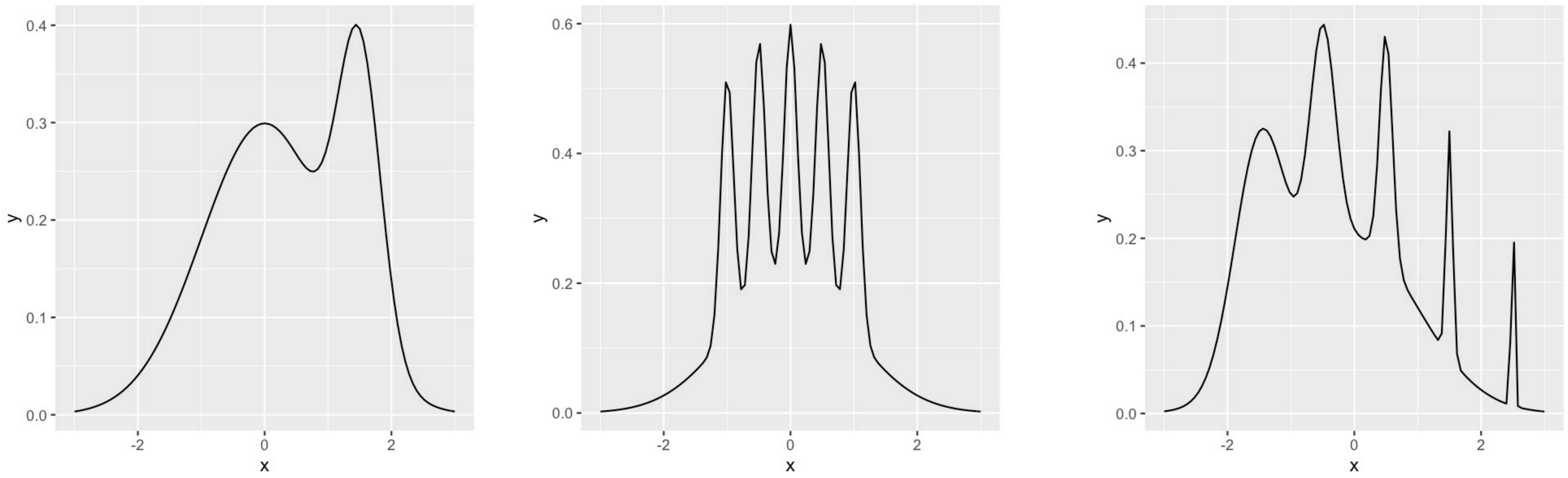


- **Universal approximation:** Gaussian mixture can approximate almost any smooth density functions arbitrarily well

Picture credit: Geoffrey McLachlan and David Peel — Finite Mixture Models

Background: Gaussian mixture

- Finite Gaussian mixture density: a **convex combination** of **finitely** many **distinct** Gaussian densities



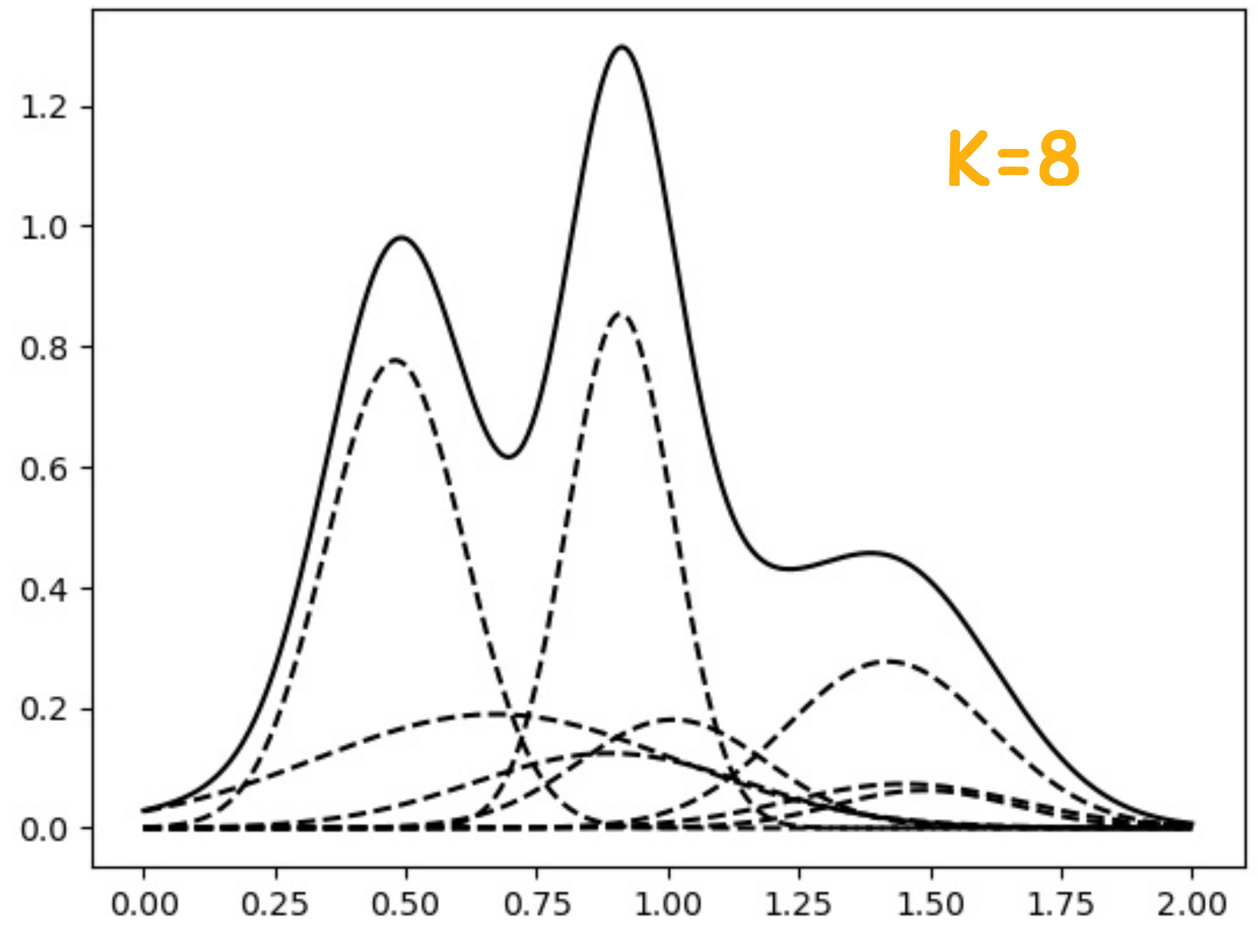
- **Universal approximation:** Gaussian mixture can approximate almost any smooth density functions arbitrarily well

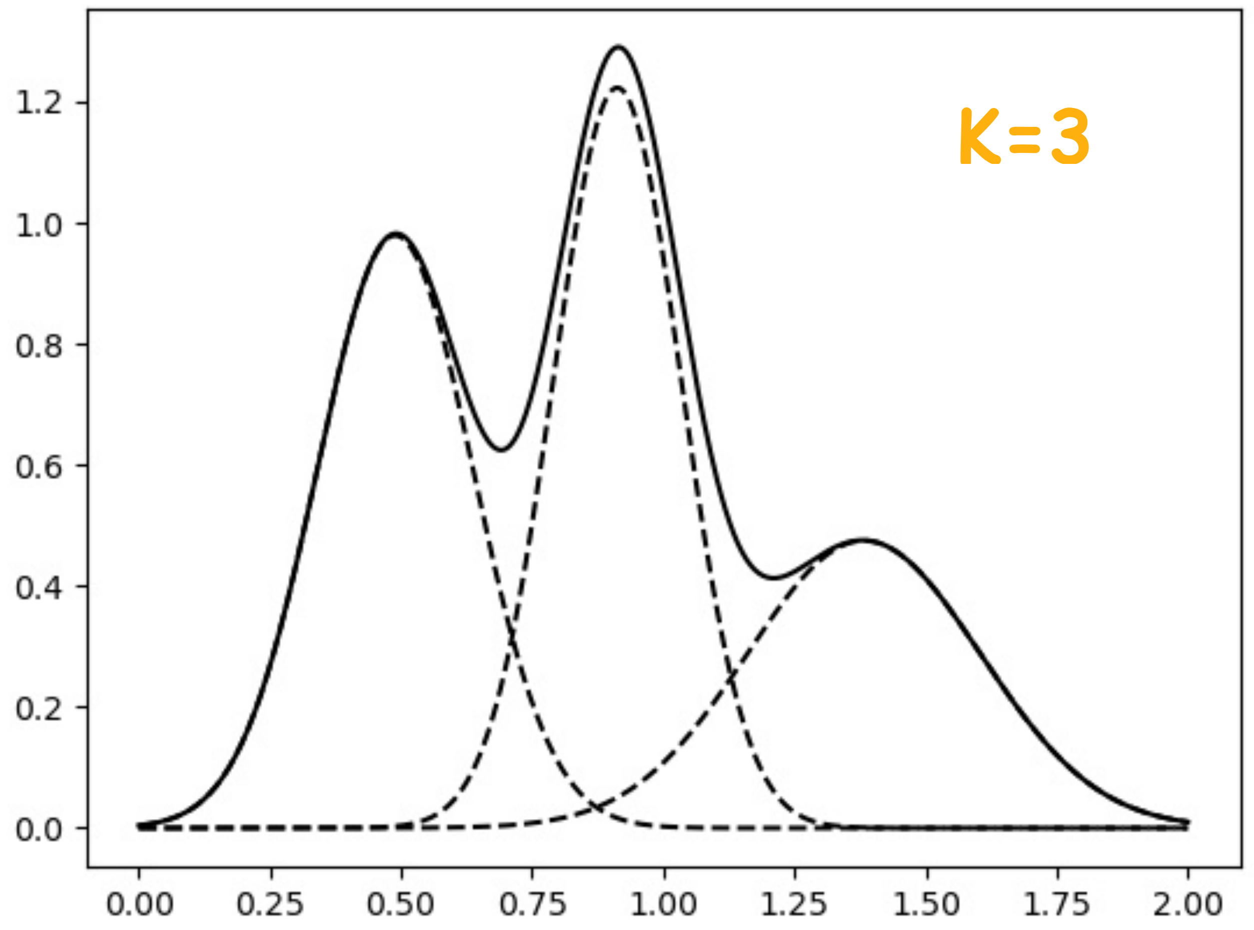
- **Application**: parametric density approximation

Picture credit: Geoffrey McLachlan and David Peel — Finite Mixture Models

What is Gaussian mixture reduction (GMR)?

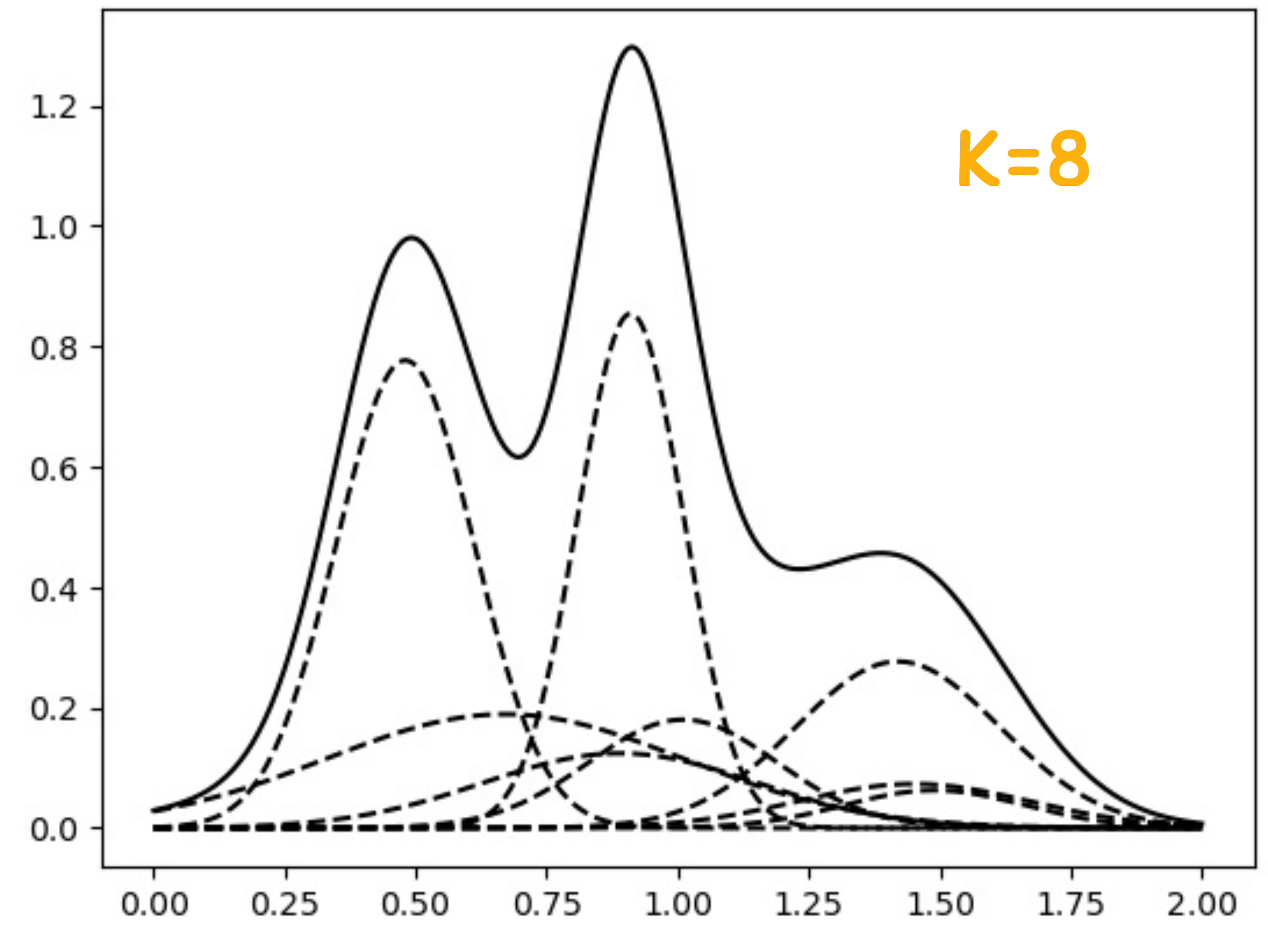
- Densities of mixtures with **different orders** may have **close shapes**

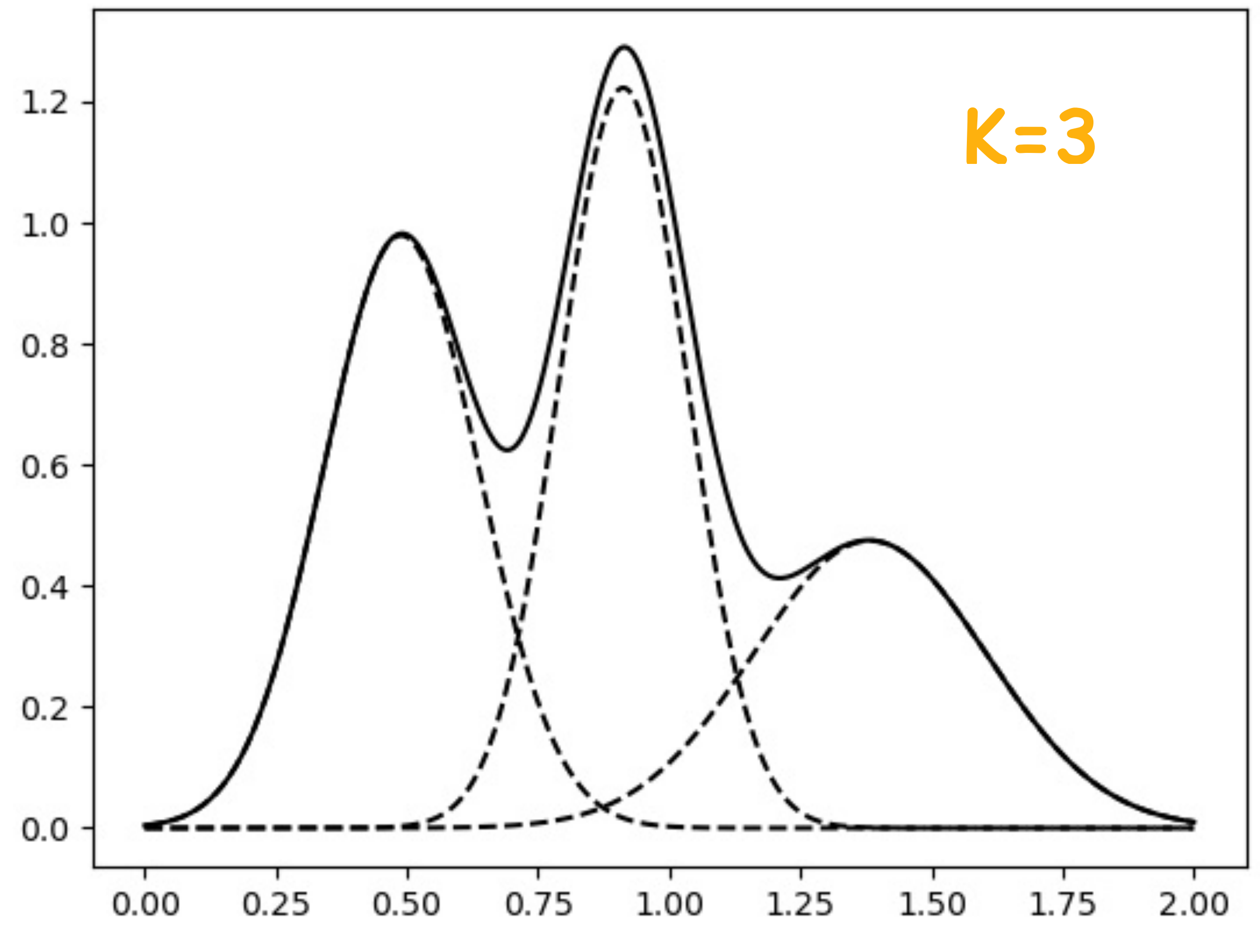




What is Gaussian mixture reduction (GMR)?

- Densities of mixtures with **different orders** may have **close shapes**

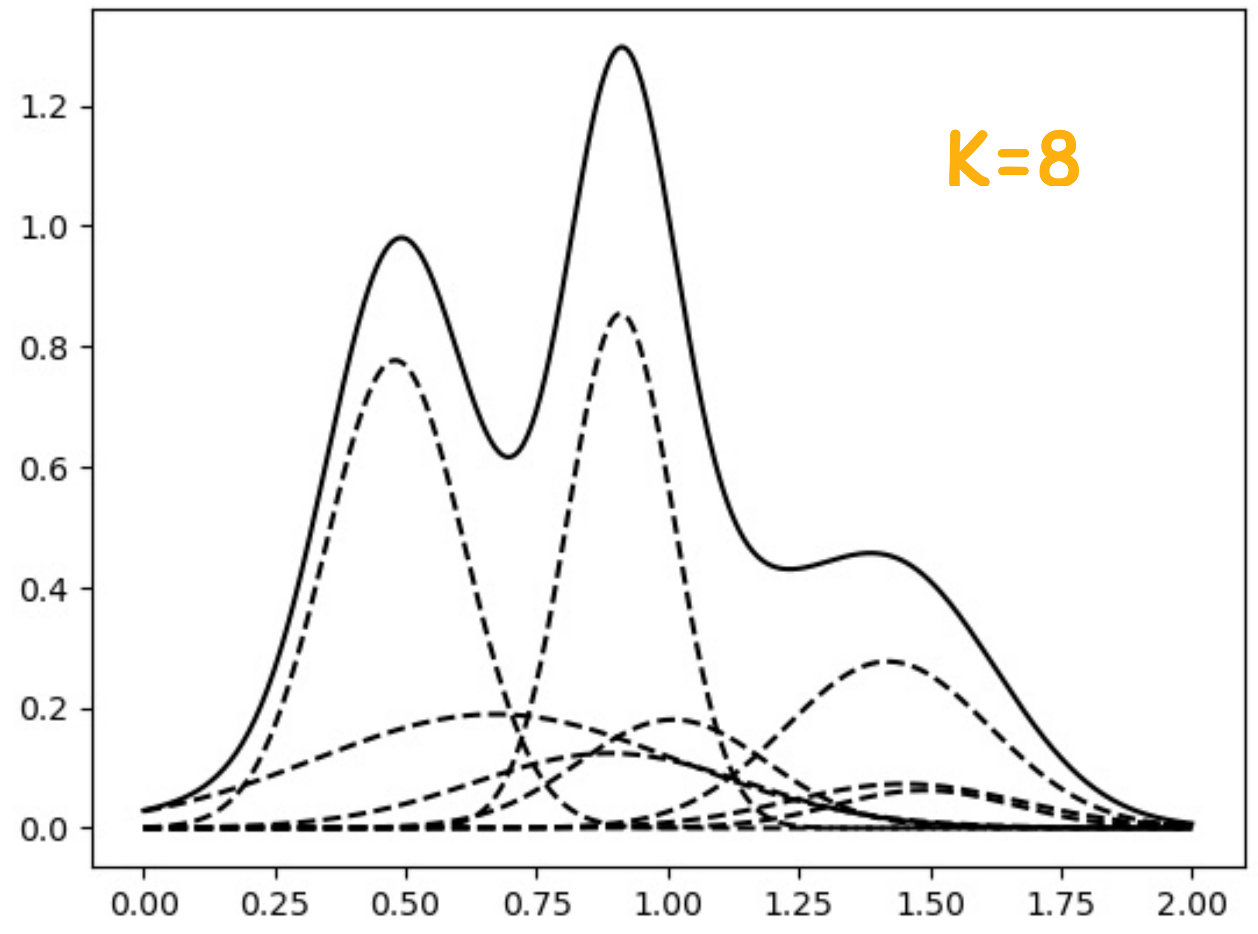


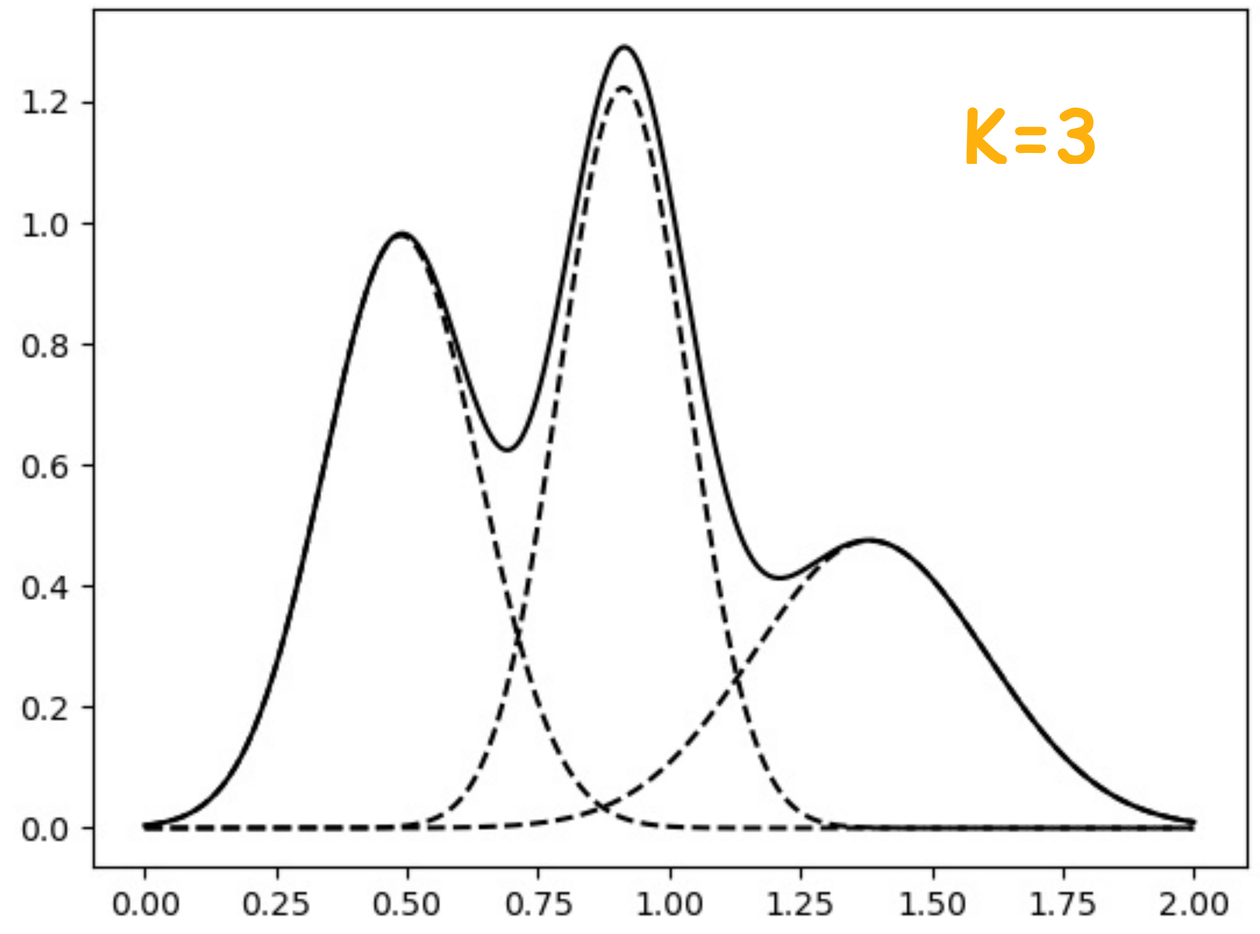


- **Gaussian mixture reduction**: approximate a high order mixture by one with a lower order

What is Gaussian mixture reduction (GMR)?

- Densities of mixtures with **different orders** may have **close shapes**



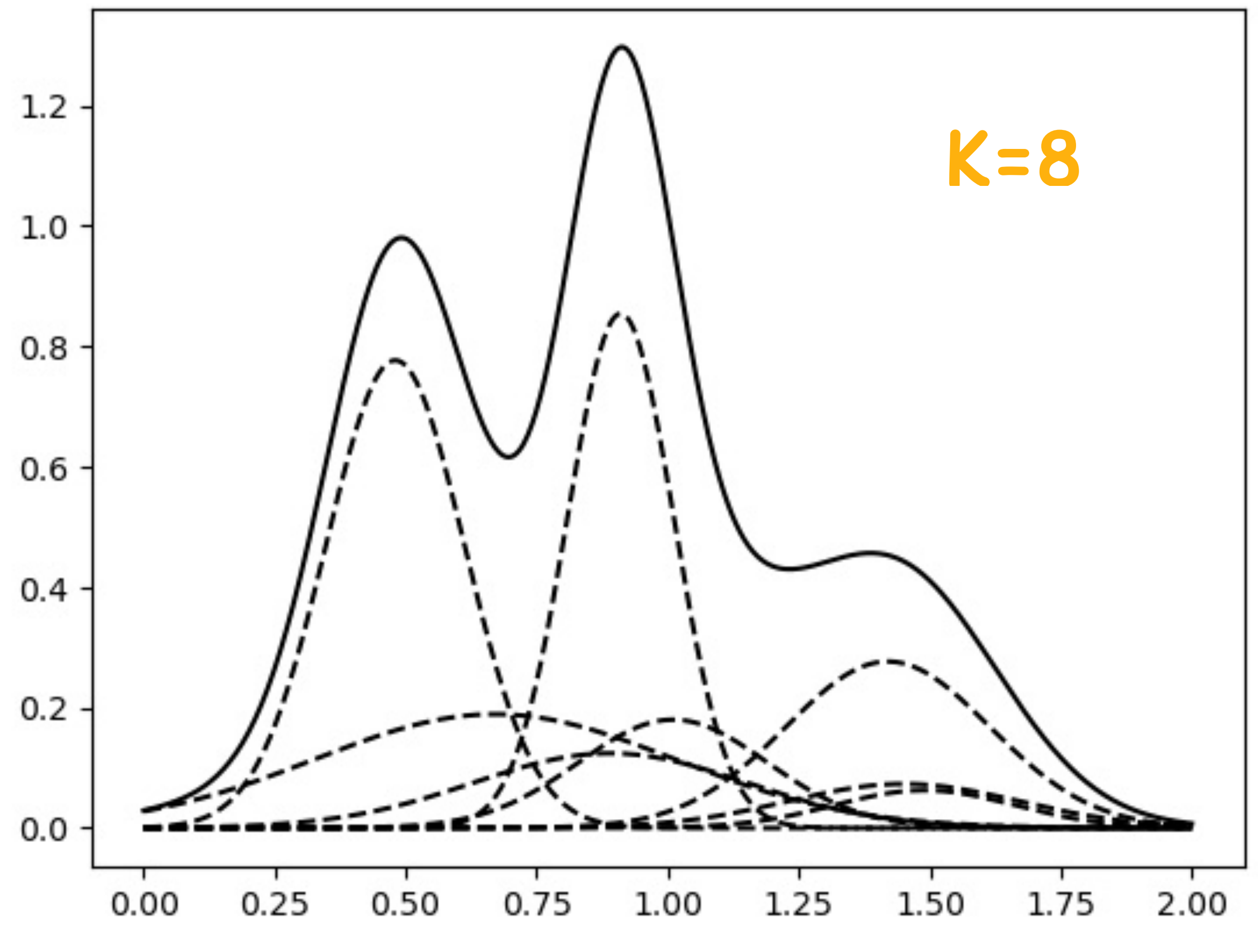


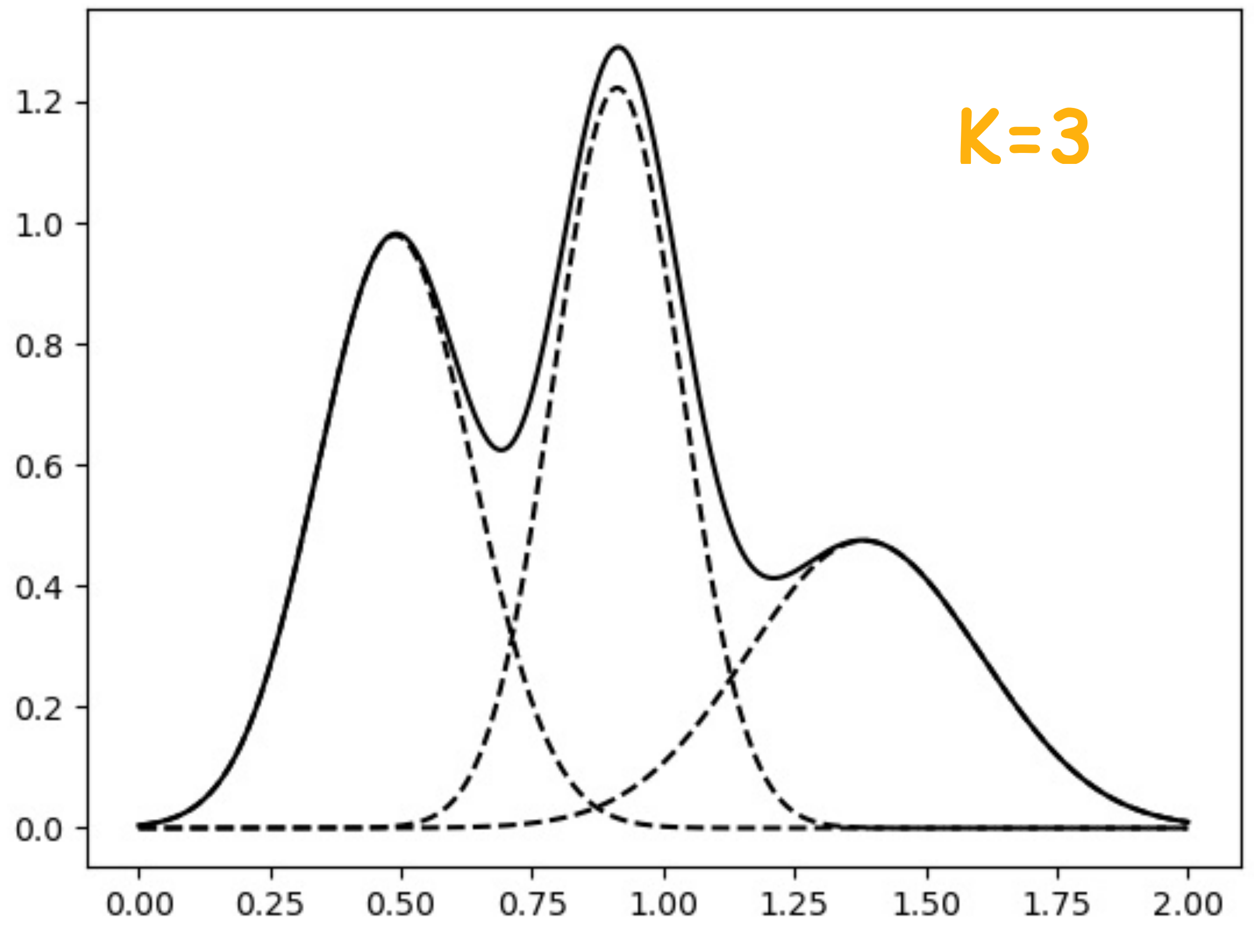
Original mixture $\phi(x; G) = \sum_{n=1}^{N} w_n \phi(x; \theta_n)$

- **Gaussian mixture reduction**: approximate a high order mixture by one with a lower order

What is Gaussian mixture reduction (GMR)?

- Densities of mixtures with **different orders** may have **close shapes**





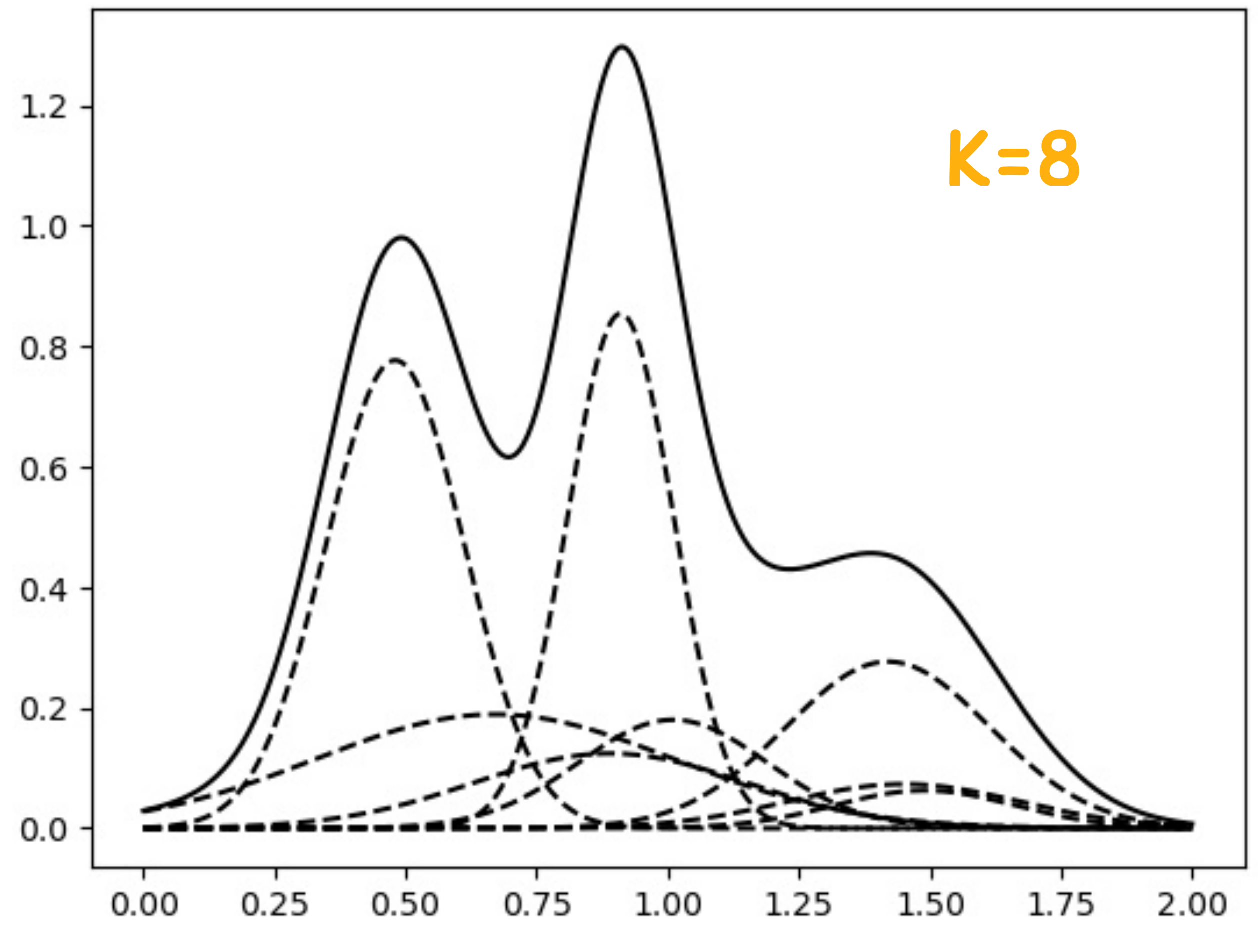
Original mixture $\phi(x; G) = \sum_{n=1}^{N} w_n \phi(x; \theta_n)$

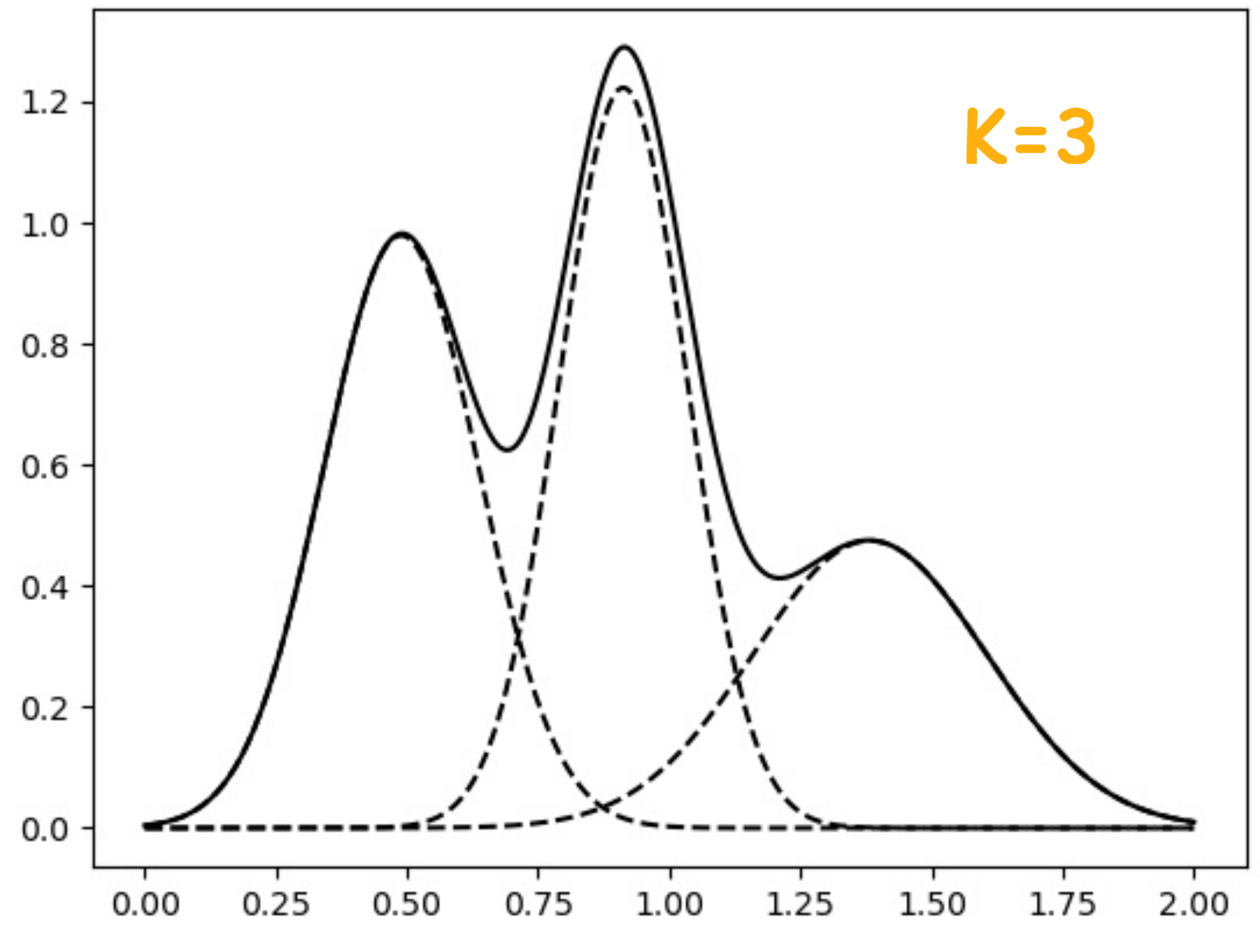
$\phi(x; \tilde{G}) = \sum_{m=1}^{M} \tilde{w}_m \phi(x; \tilde{\theta}_m)$ Reduced mixture

- **Gaussian mixture reduction**: approximate a high order mixture by one with a lower order

What is Gaussian mixture reduction (GMR)?

- Densities of mixtures with **different orders** may have **close shapes**





Original mixture $$\phi(x;G) = \sum_{n=1}^{N} w_n \phi(x;\theta_n)$$ $\approx$ $M \ll N$ $$\phi(x;\tilde{G}) = \sum_{m=1}^{M} \tilde{w}_m \phi(x;\tilde{\theta}_m)$$ Reduced mixture

- **Gaussian mixture reduction:** approximate a high order mixture by one with a lower order

Why GMR?

- Higher order mixture → Heavier downstream computation cost

Why GMR?

- Higher order mixture → Heavier downstream computation cost
- Orders does not carry scientific meanings in approximation

Why GMR?

- Higher order mixture → Heavier downstream computation cost
- Orders does not carry scientific meanings in approximation
- **Applications**

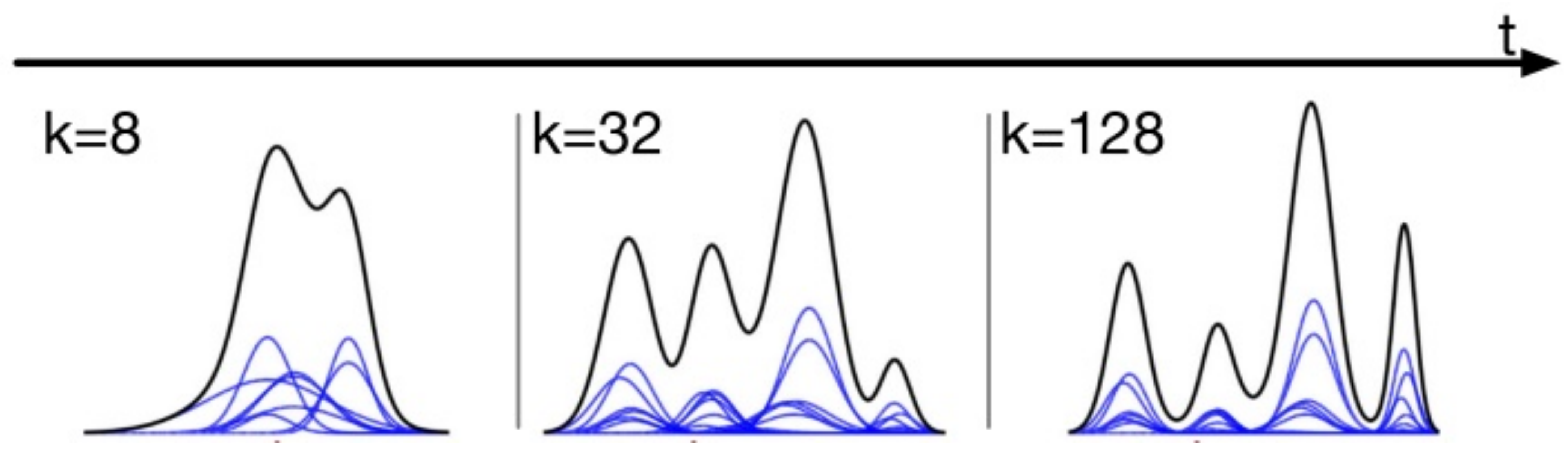


Figure credit: Lei Yu et al. 2018

Recursive inference

- Belief propagation in graphical model (Yu et al., 2018)
- Tracking in hidden Markov model (Brubaker et al., 2015)

Why GMR?

- Higher order mixture → Heavier downstream computation cost
- Orders does not carry scientific meanings in approximation
- **Applications**

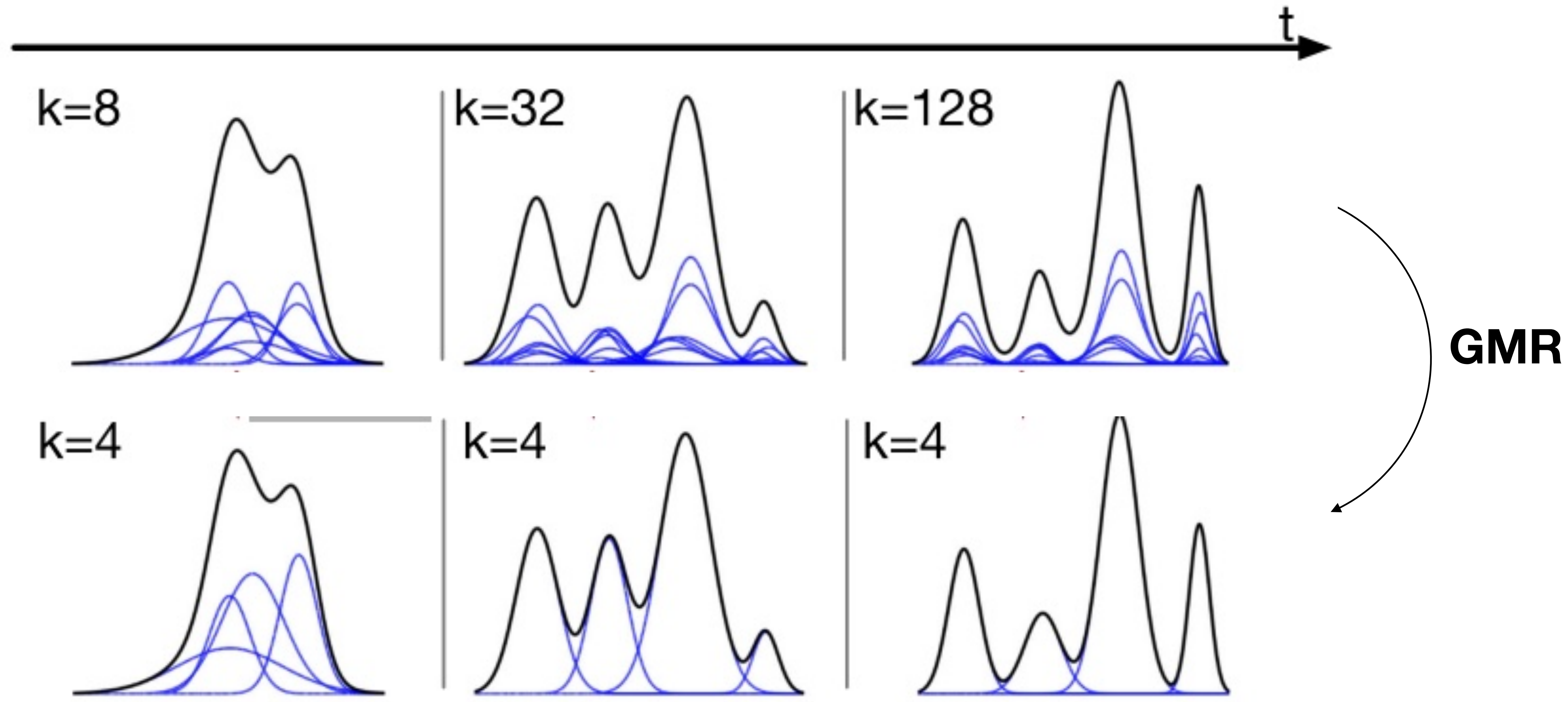


Figure credit: Lei Yu et al. 2018

Recursive inference

- Belief propagation in graphical model (Yu et al., 2018)
- Tracking in hidden Markov model (Brubaker et al., 2015)

Why GMR?

- Higher order mixture → Heavier downstream computation cost
- Orders does not carry scientific meanings in approximation
- **Applications**

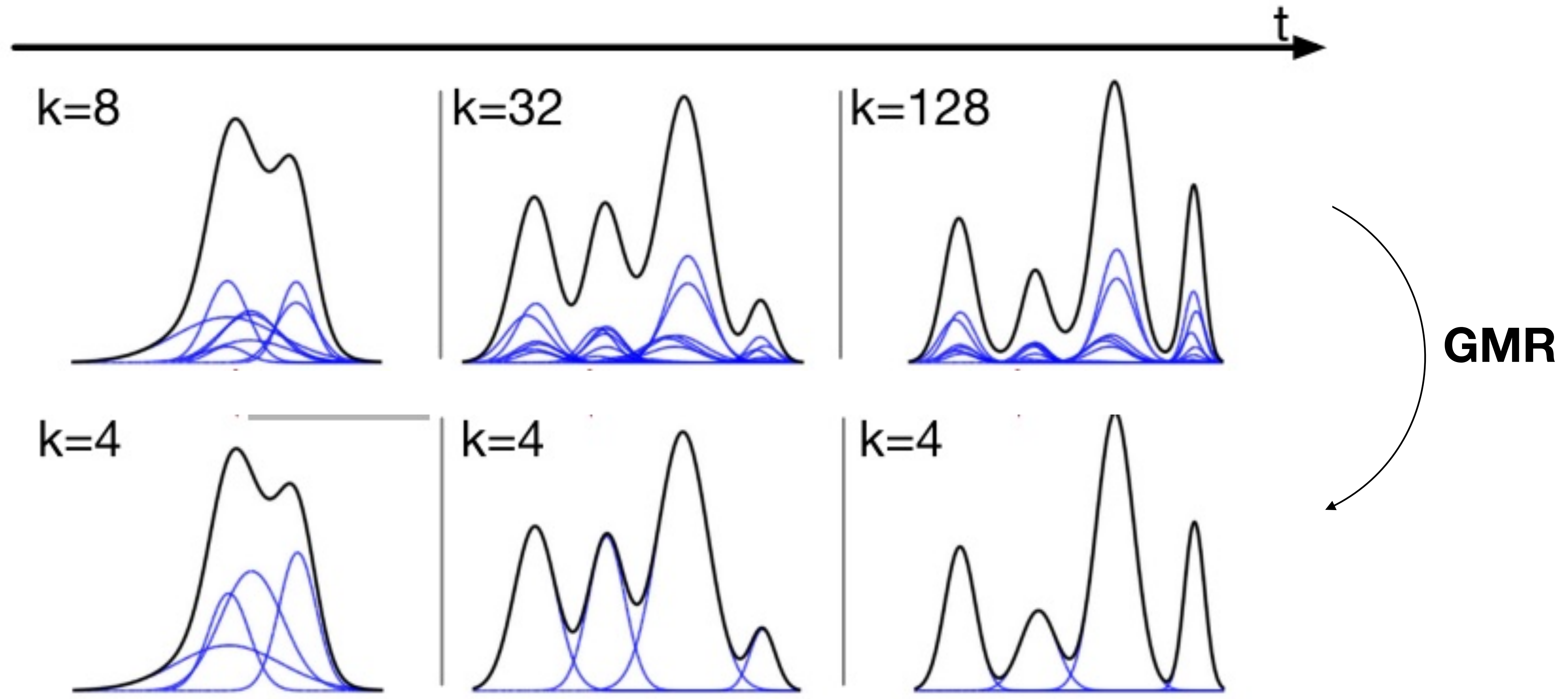
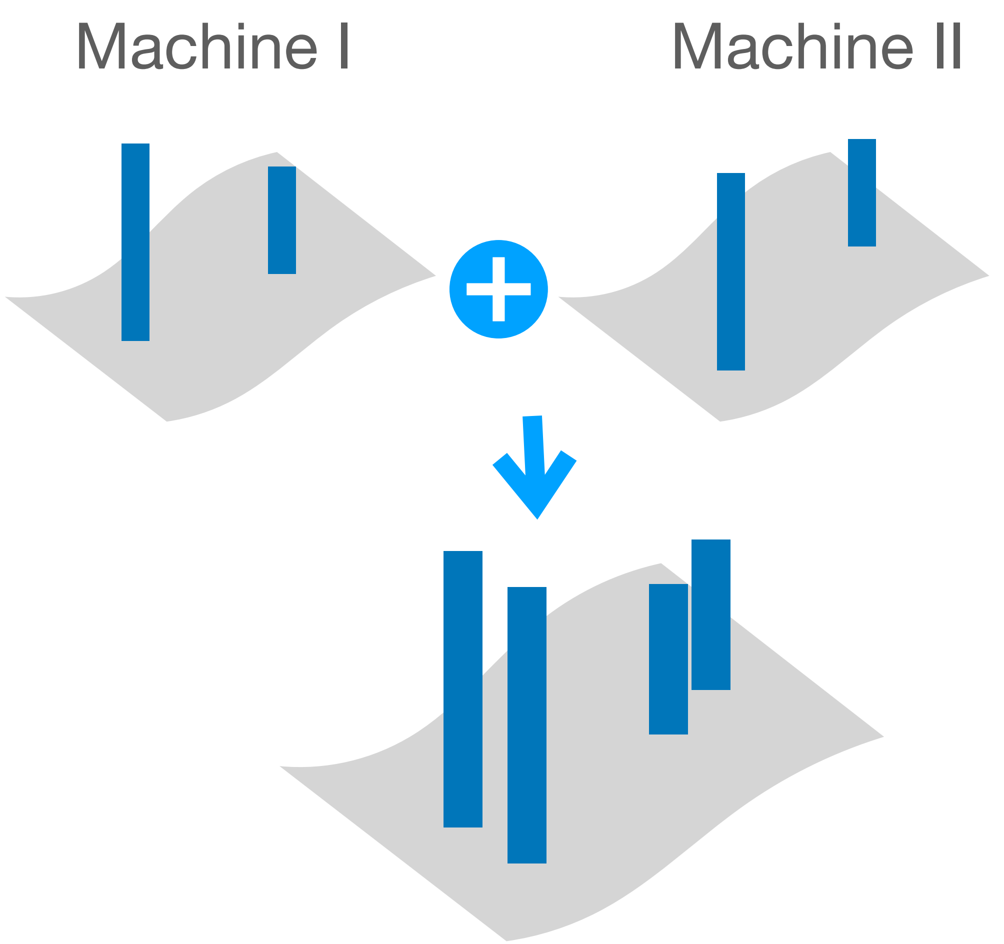


Figure credit: Lei Yu et al. 2018

Recursive inference

- Belief propagation in graphical model (Yu et al., 2018)
- Tracking in hidden Markov model (Brubaker et al., 2015)



Distributed learning (Zhang & Chen 2022)

Why GMR?

- Higher order mixture → Heavier downstream computation cost
- Orders does not carry scientific meanings in approximation
- **Applications**

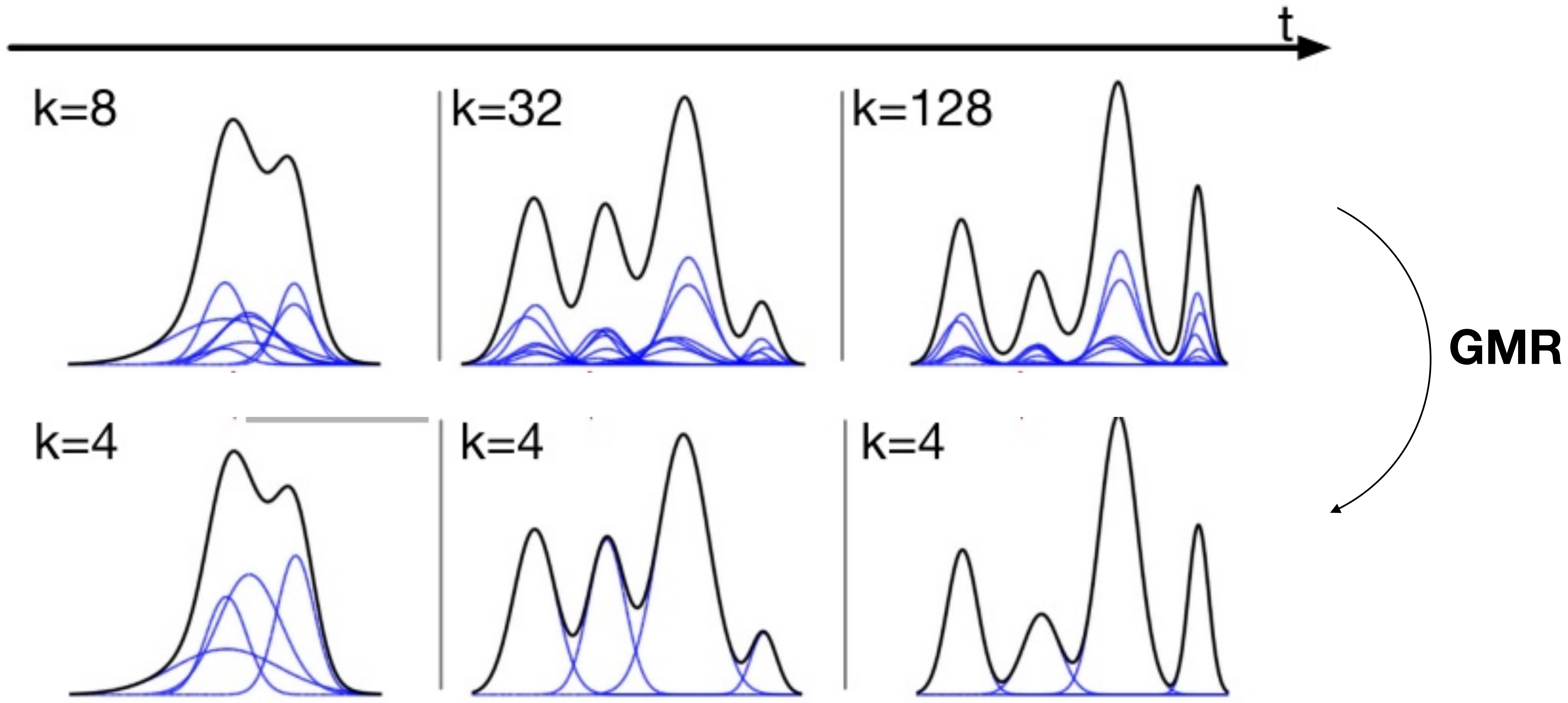
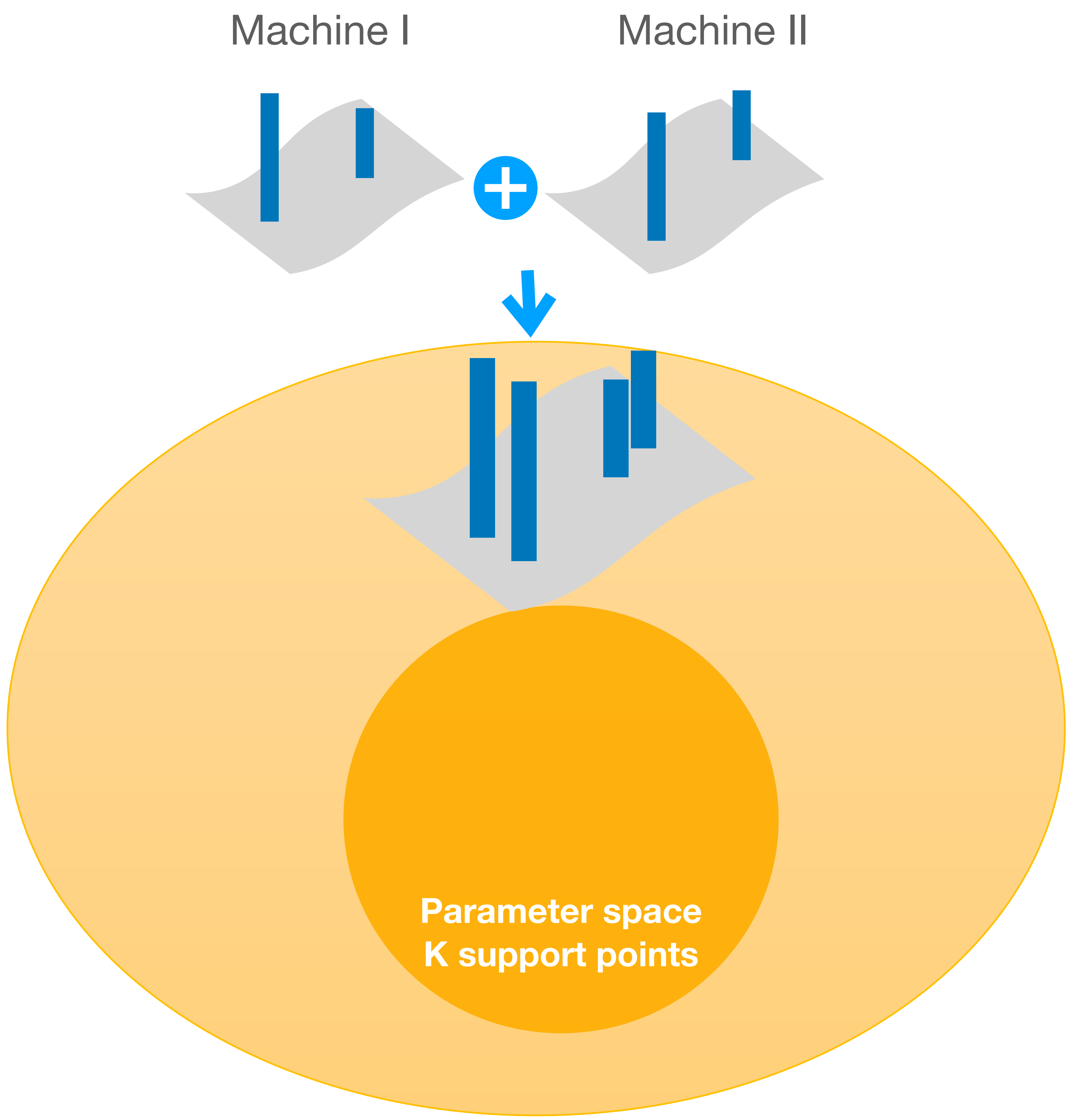


Figure credit: Lei Yu et al. 2018

Recursive inference

- Belief propagation in graphical model (Yu et al., 2018)
- Tracking in hidden Markov model (Brubaker et al., 2015)



Distributed learning (Zhang & Chen 2022)

Why GMR?

- Higher order mixture → Heavier downstream computation cost
- Orders does not carry scientific meanings in approximation
- **Applications**

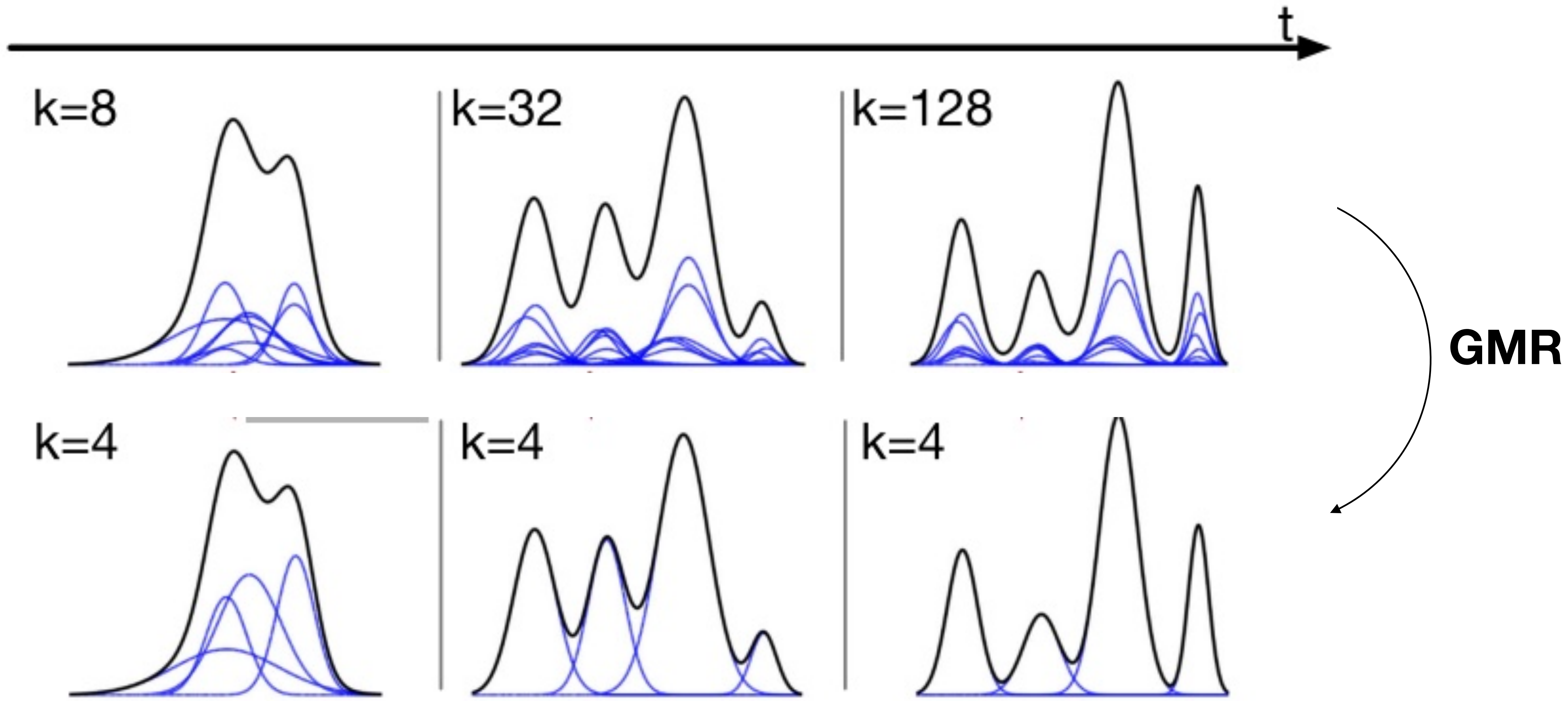
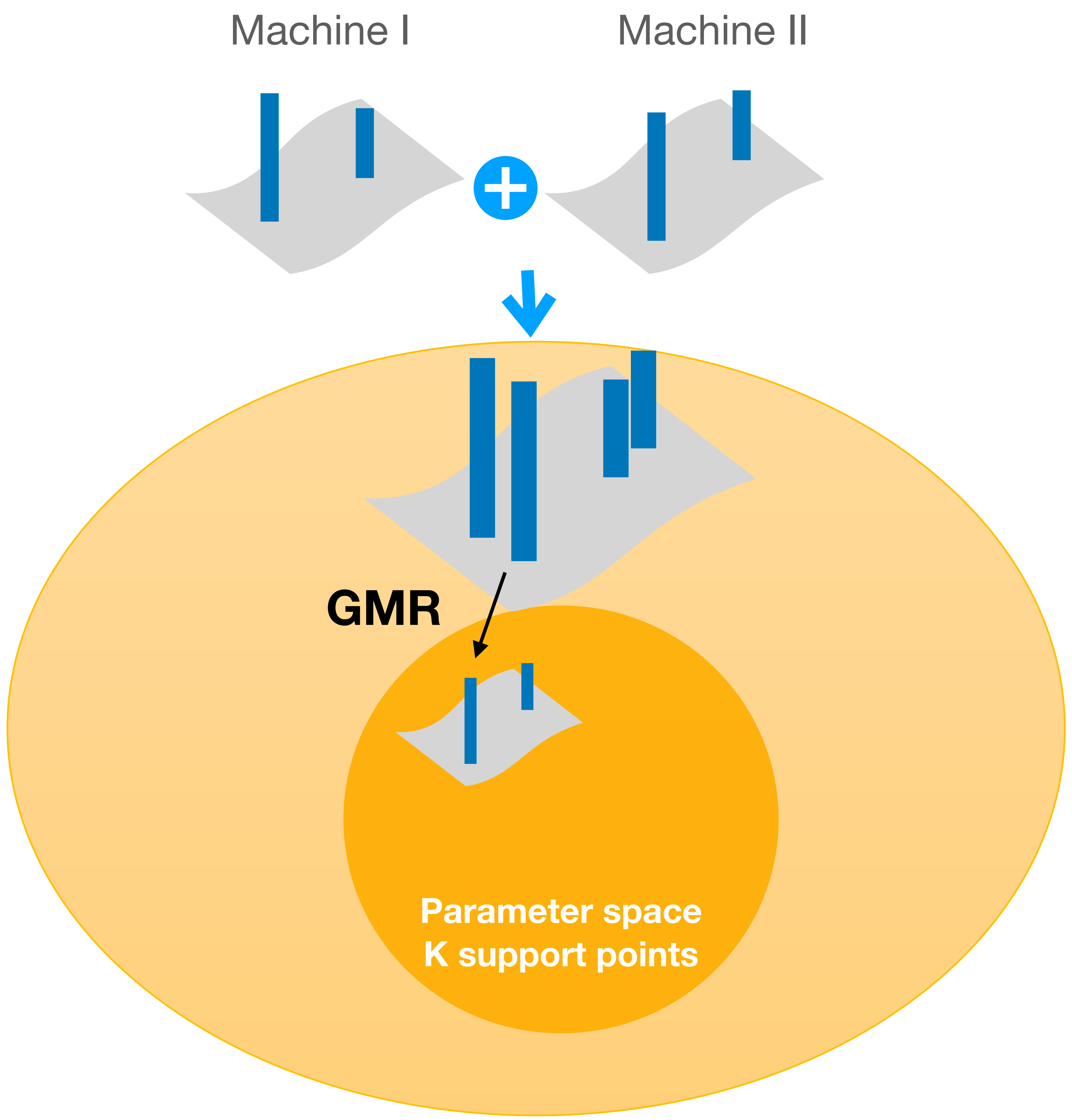


Figure credit: Lei Yu et al. 2018



Recursive inference

- Belief propagation in graphical model (Yu et al., 2018)
- Tracking in hidden Markov model (Brubaker et al., 2015)

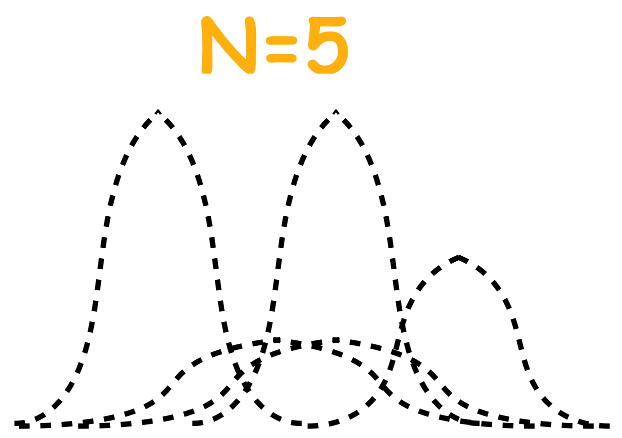
Distributed learning (Zhang & Chen 2022)

Existing approaches

- **Greedy algorithm** (*Salmond, 1990; Runnalls, 2007; Assa and Plataniotis, 2018*)

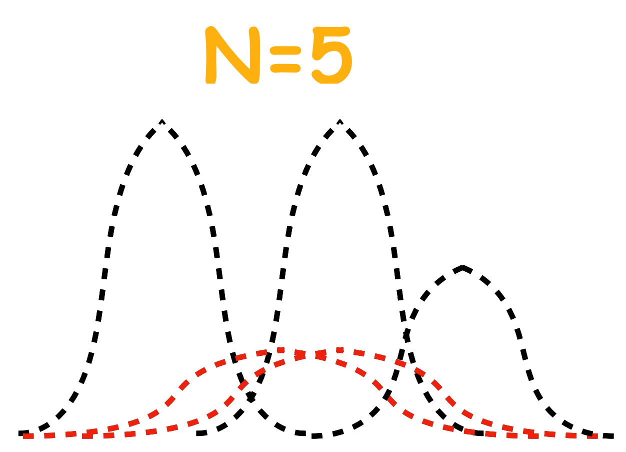
Existing approaches

- **Greedy algorithm** (*Salmond, 1990; Runnalls, 2007; Assa and Plataniotis, 2018*)



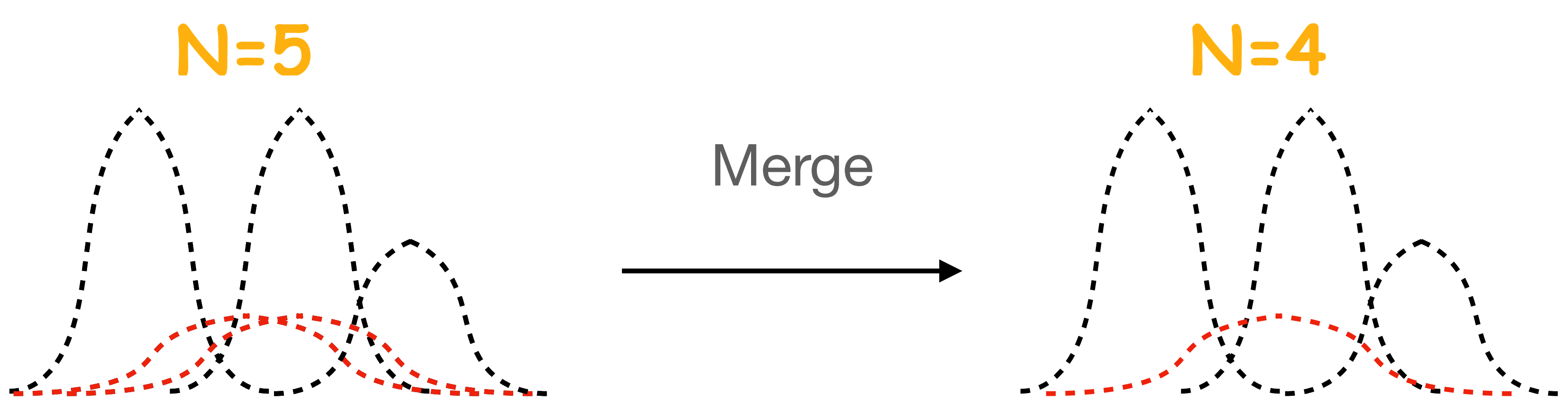
Existing approaches

- **Greedy algorithm** (*Salmond, 1990; Runnalls, 2007; Assa and Plataniotis, 2018*)



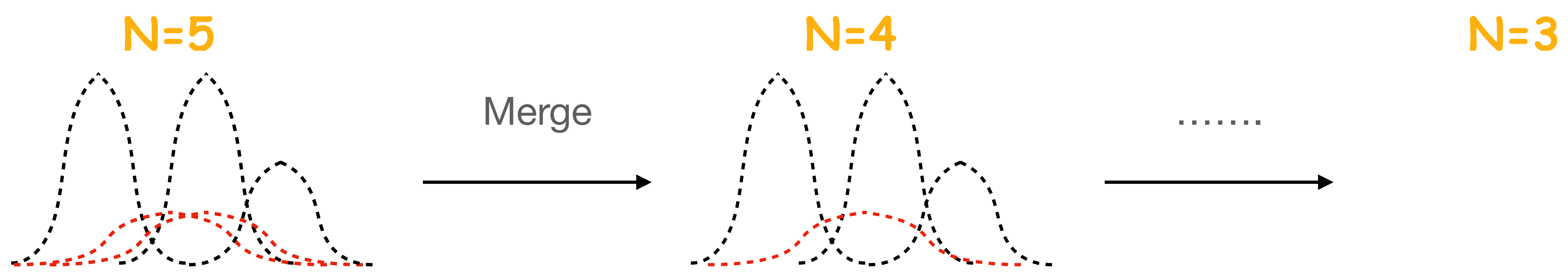
Existing approaches

- **Greedy algorithm** (*Salmond, 1990; Runnalls, 2007; Assa and Plataniotis, 2018*)



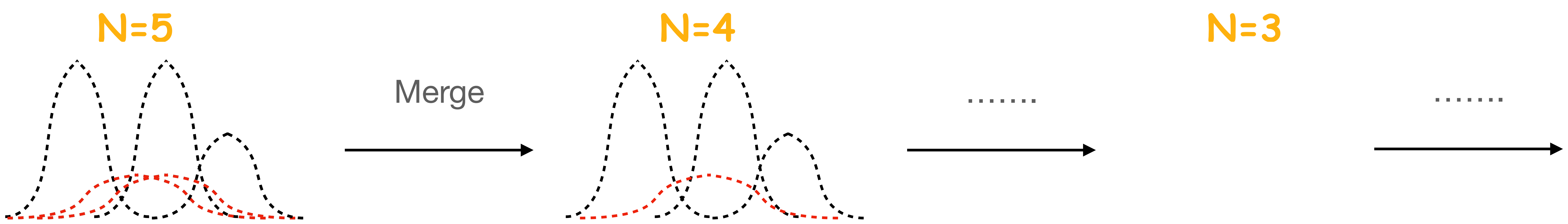
Existing approaches

- **Greedy algorithm** (*Salmond, 1990; Runnalls, 2007; Assa and Plataniotis, 2018*)



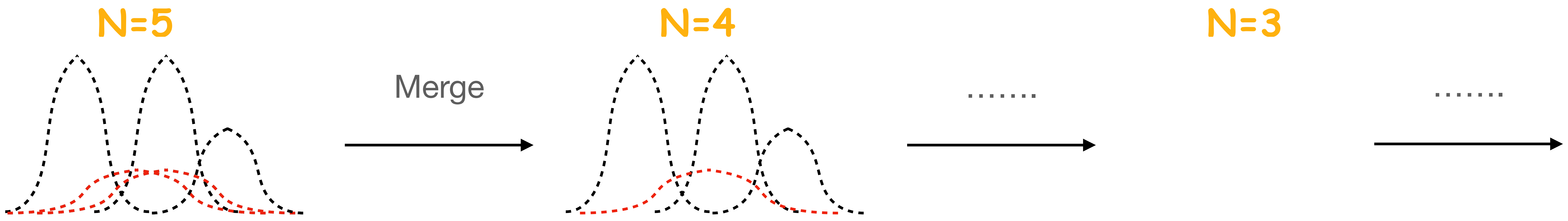
Existing approaches

- **Greedy algorithm** (*Salmond, 1990; Runnalls, 2007; Assa and Plataniotis, 2018*)



Existing approaches

- **Greedy algorithm** (*Salmond, 1990; Runnalls, 2007; Assa and Plataniotis, 2018*)

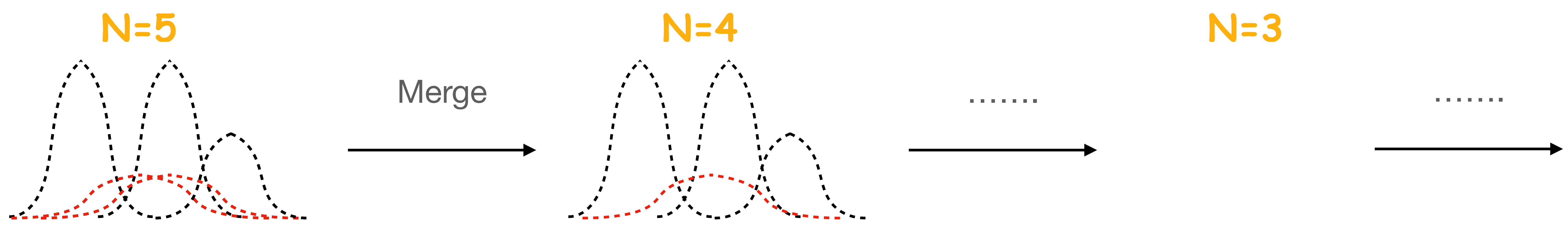


- **Optimization-based** (*Williams and Maybeck, 2006*): directly search for

$$\tilde{G} = \text{argmin}_{G^{\dagger} \in \mathbb{G}_M} \int \{\phi(x; G) - \phi(x; G^{\dagger})\}^2 dx$$

Existing approaches

- **Greedy algorithm** (*Salmond, 1990; Runnalls, 2007; Assa and Plataniotis, 2018*)



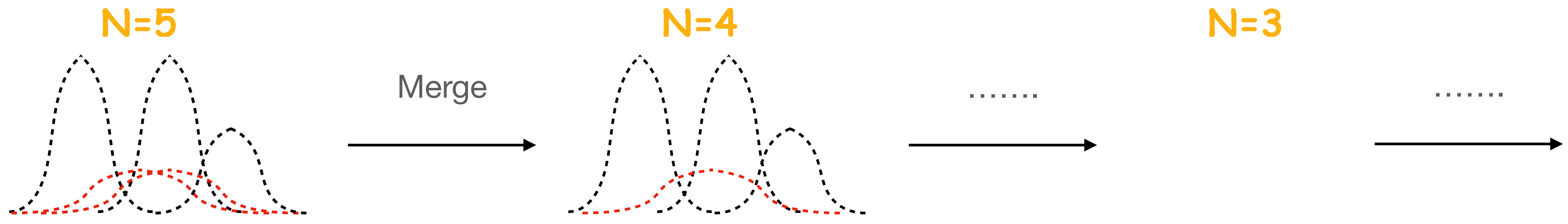
- **Optimization-based** (*Williams and Maybeck, 2006*): directly search for

$$\tilde{G} = \text{argmin}_{G^\dagger \in \mathbb{G}_M} \int \{\phi(x; G) - \phi(x; G^\dagger)\}^2 dx$$

- **Clustering-based** (*Schieferdecker and Huber, 2009; Assa and Plataniotis, 2018*)

Existing approaches

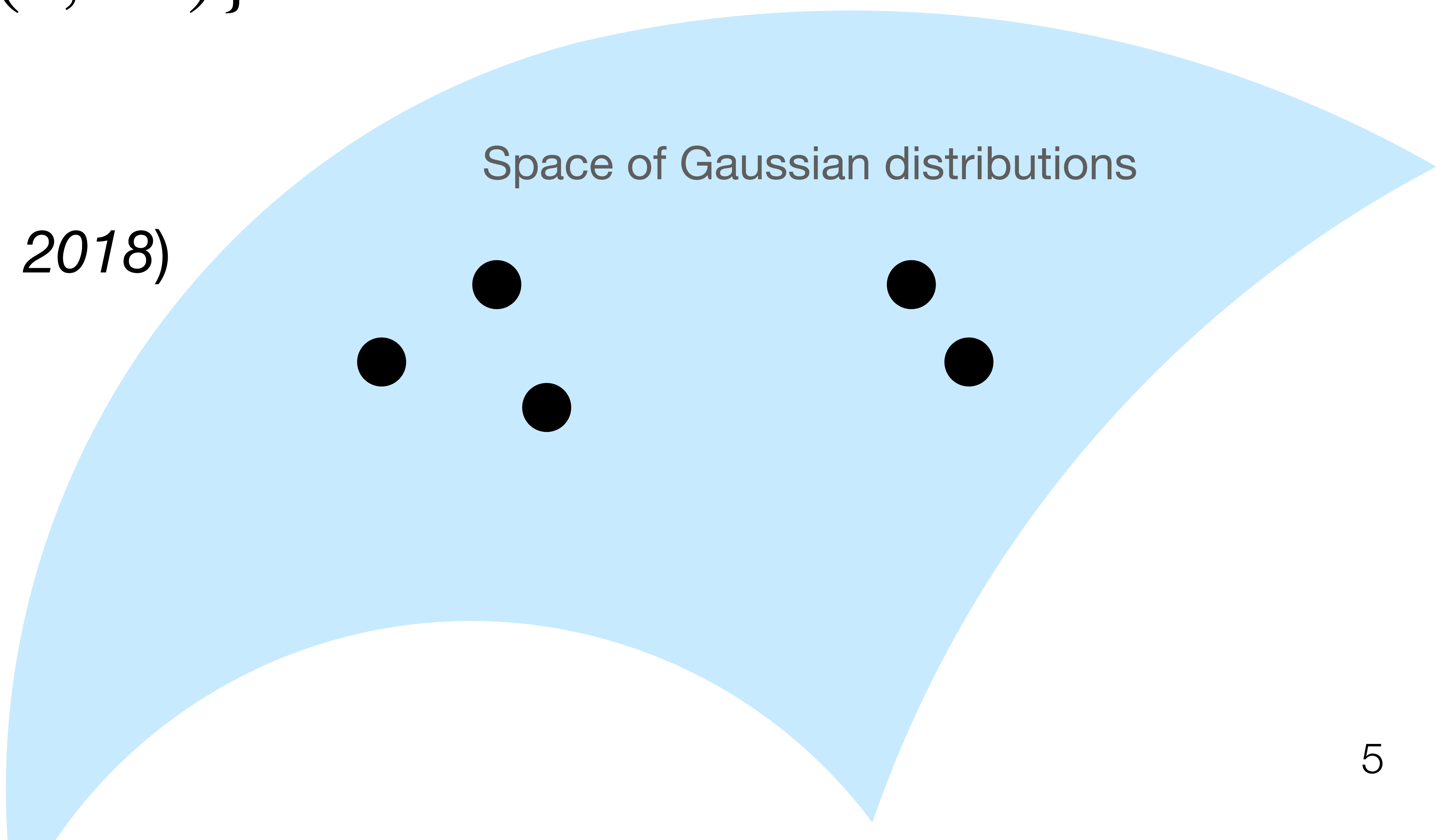
- **Greedy algorithm** (*Salmond, 1990; Runnalls, 2007; Assa and Plataniotis, 2018*)



- **Optimization-based** (*Williams and Maybeck, 2006*): directly search for

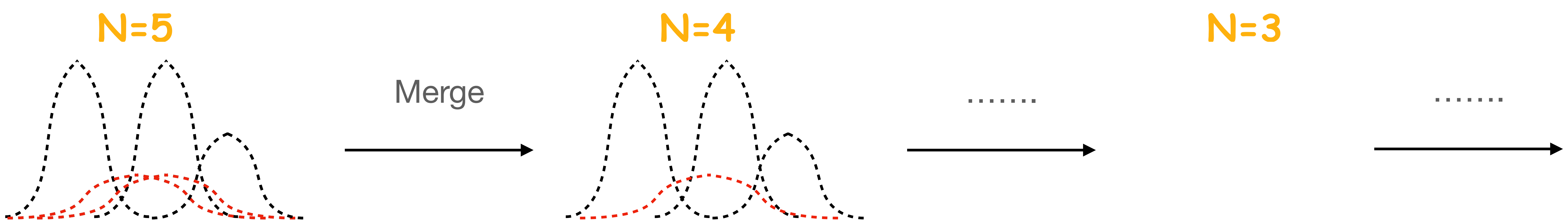
$$\tilde{G} = \text{argmin}_{G^\dagger \in \mathbb{G}_M} \int \{\phi(x; G) - \phi(x; G^\dagger)\}^2 dx$$

- **Clustering-based** (*Schieferdecker and Huber, 2009; Assa and Plataniotis, 2018*)



Existing approaches

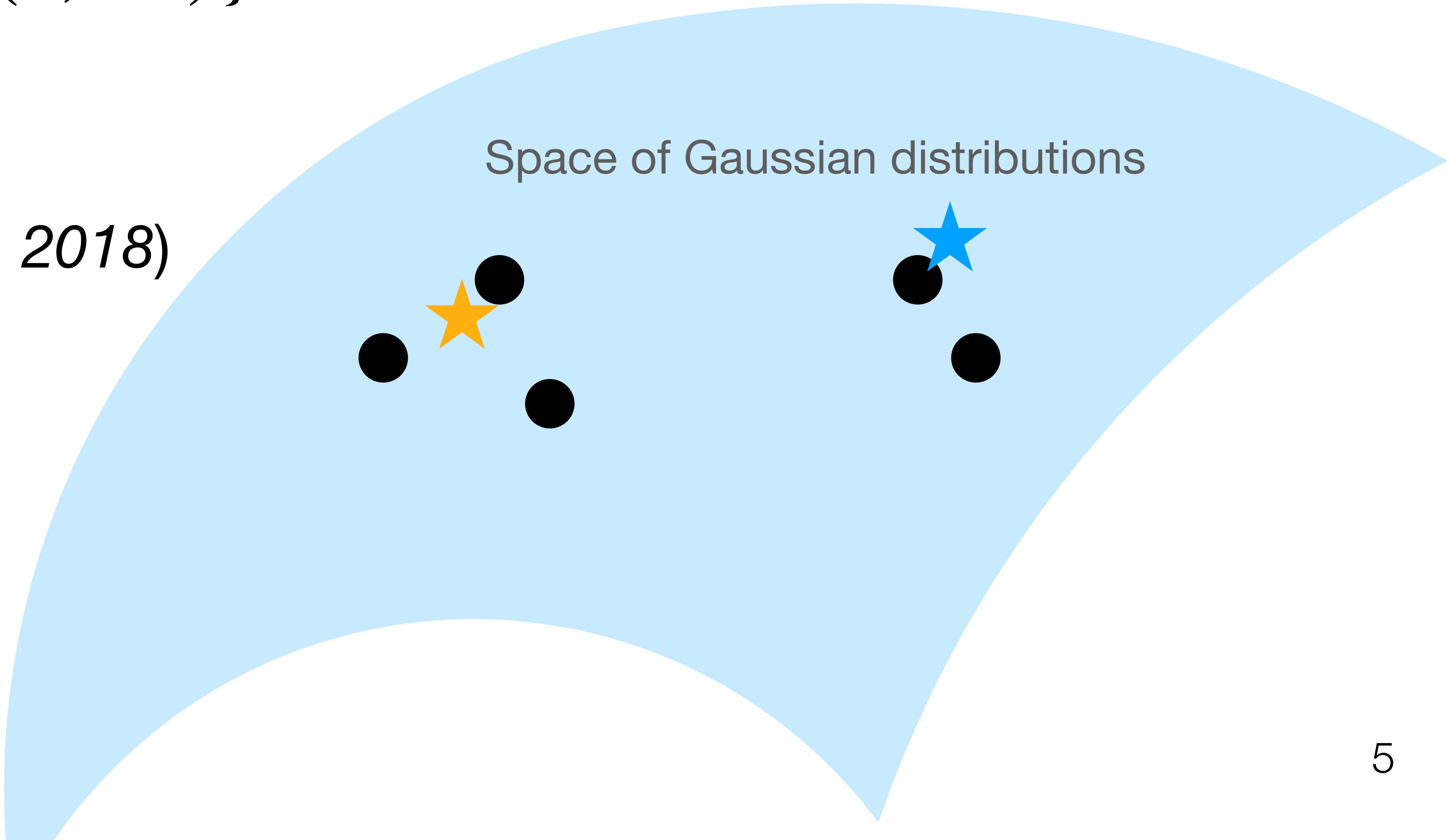
- **Greedy algorithm** (*Salmond, 1990; Runnalls, 2007; Assa and Plataniotis, 2018*)



- **Optimization-based** (*Williams and Maybeck, 2006*): directly search for

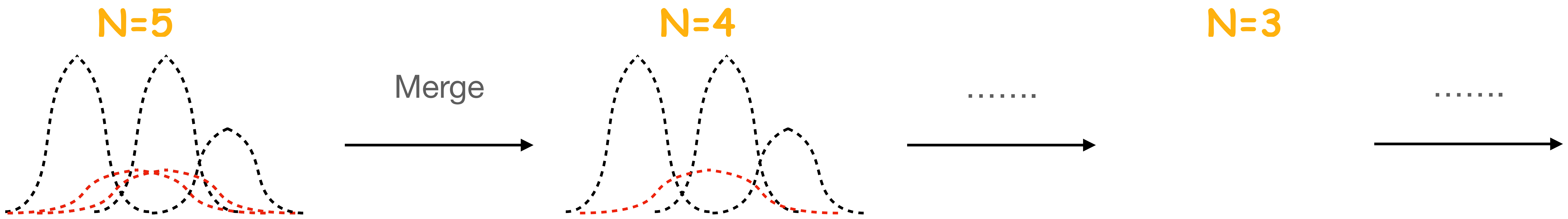
$$\tilde{G} = \text{argmin}_{G^\dagger \in \mathbb{G}_M} \int \{\phi(x; G) - \phi(x; G^\dagger)\}^2 dx$$

- **Clustering-based** (*Schieferdecker and Huber, 2009; Assa and Plataniotis, 2018*)



Existing approaches

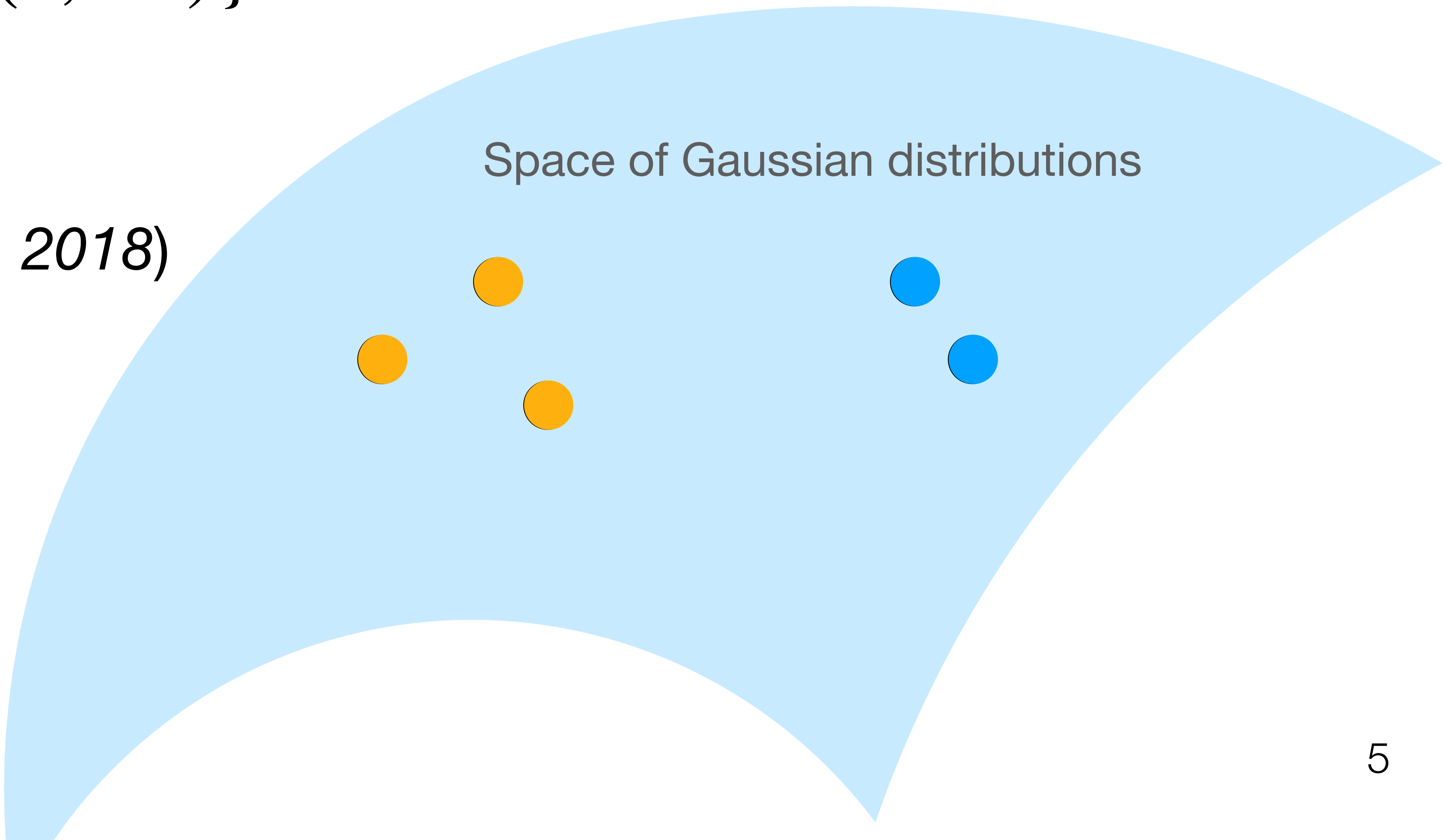
- **Greedy algorithm** (*Salmond, 1990; Runnalls, 2007; Assa and Plataniotis, 2018*)



- **Optimization-based** (*Williams and Maybeck, 2006*): directly search for

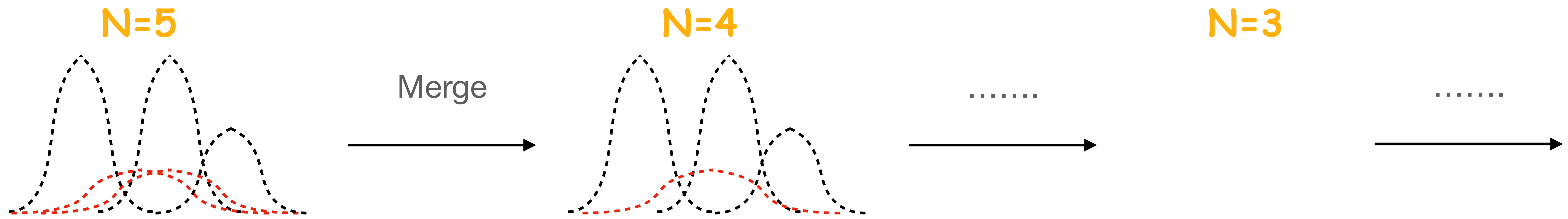
$$\tilde{G} = \text{argmin}_{G^\dagger \in \mathbb{G}_M} \int \{\phi(x; G) - \phi(x; G^\dagger)\}^2 dx$$

- **Clustering-based** (*Schieferdecker and Huber, 2009; Assa and Plataniotis, 2018*)



Existing approaches

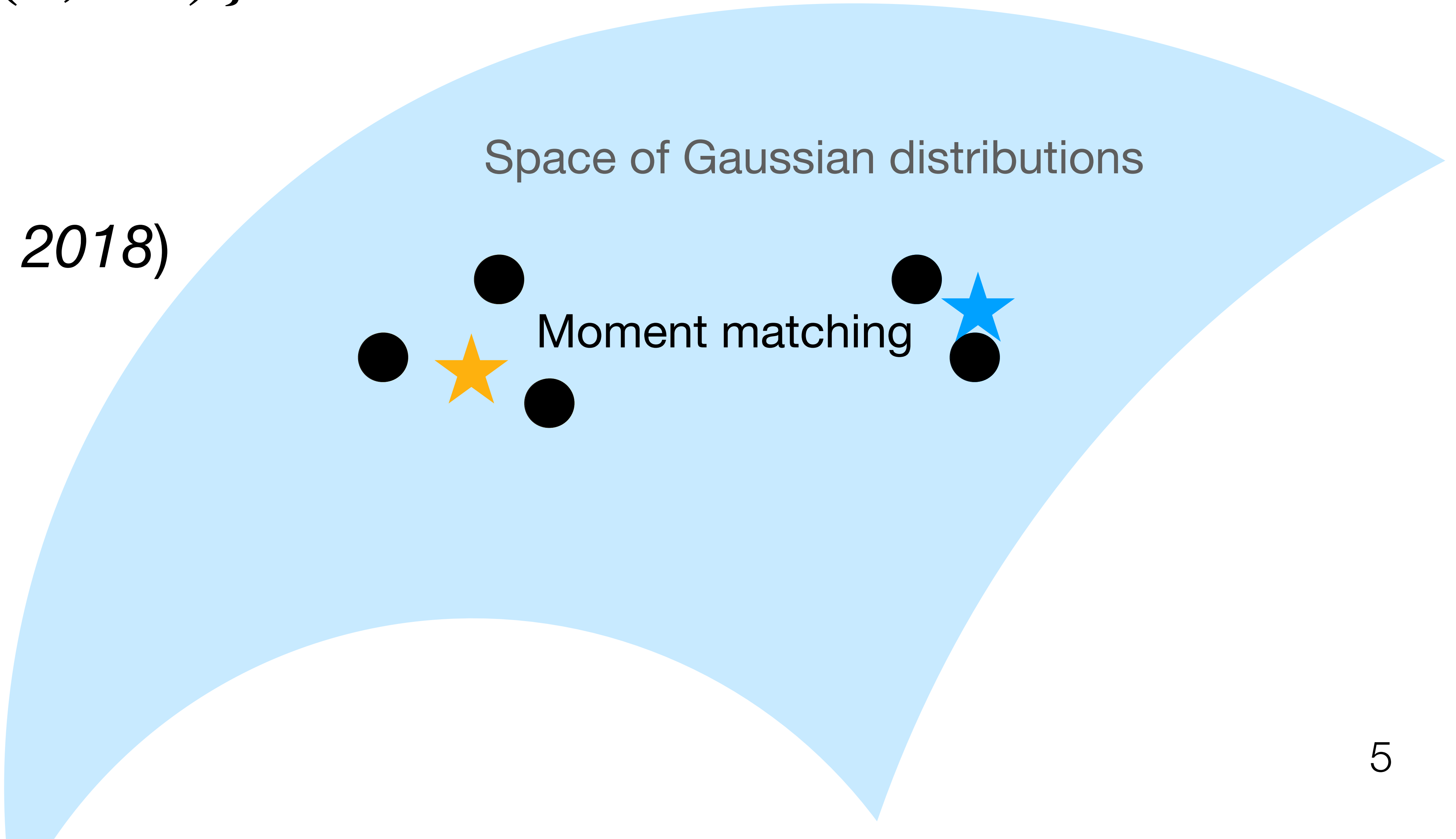
- **Greedy algorithm** (*Salmond, 1990; Runnalls, 2007; Assa and Plataniotis, 2018*)



- **Optimization-based** (*Williams and Maybeck, 2006*): directly search for

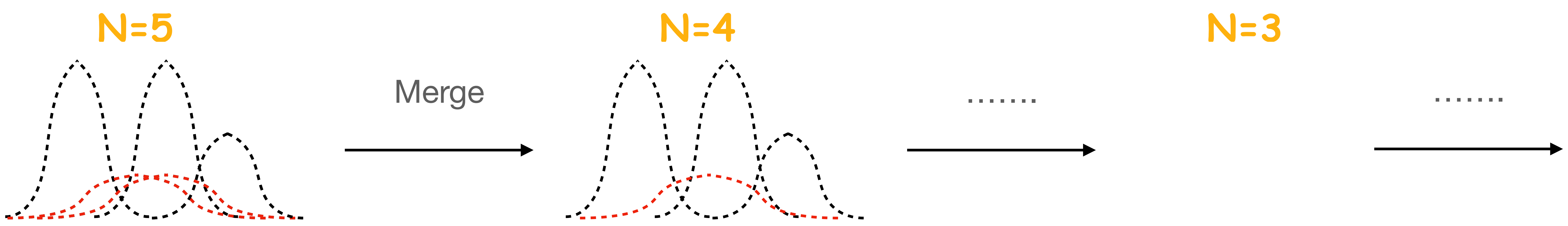
$$\tilde{G} = \text{argmin}_{G^\dagger \in \mathbb{G}_M} \int \{\phi(x; G) - \phi(x; G^\dagger)\}^2 dx$$

- **Clustering-based** (*Schieferdecker and Huber, 2009; Assa and Plataniotis, 2018*)



Existing approaches

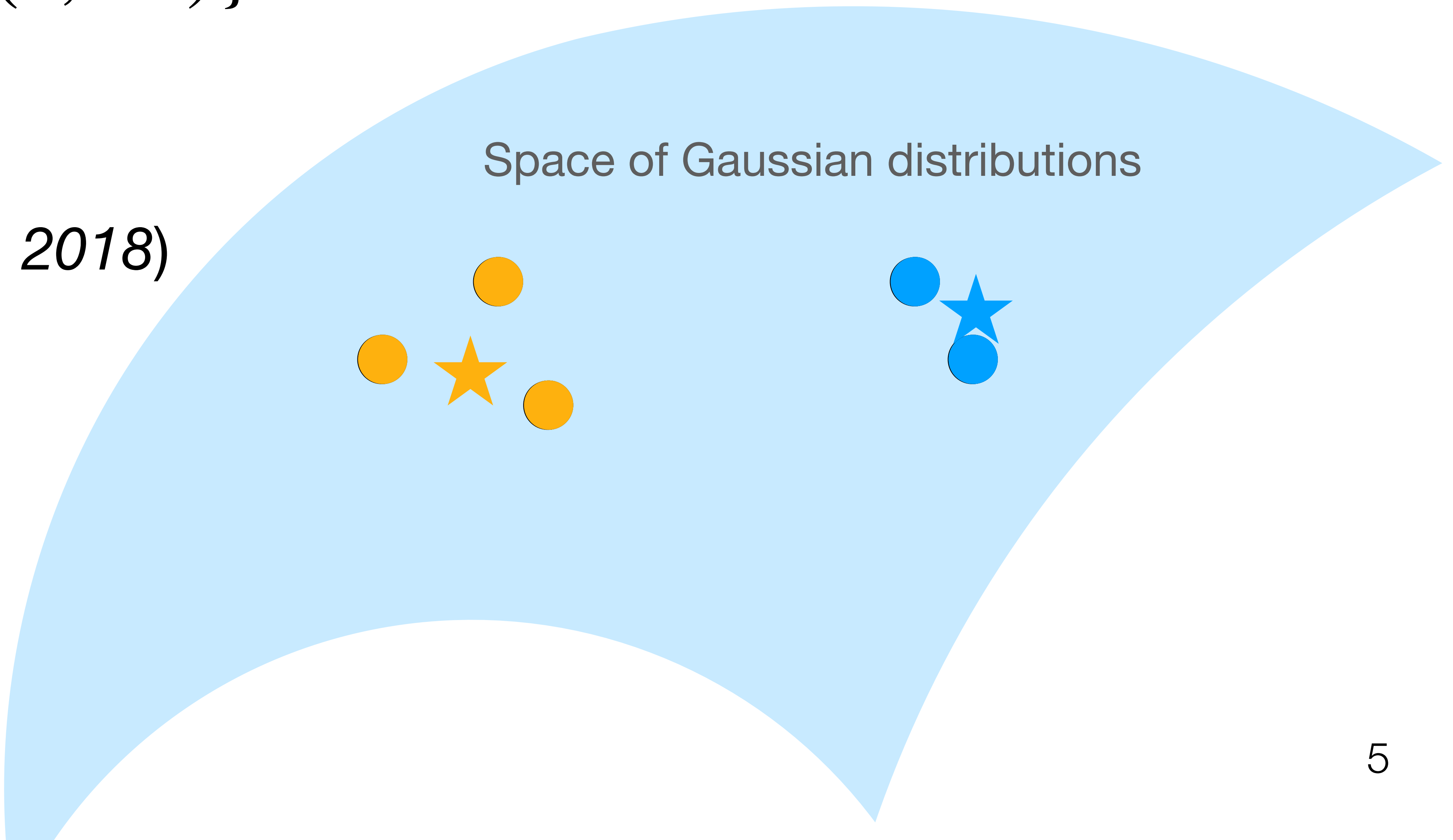
- **Greedy algorithm** (*Salmond, 1990; Runnalls, 2007; Assa and Plataniotis, 2018*)



- **Optimization-based** (*Williams and Maybeck, 2006*): directly search for

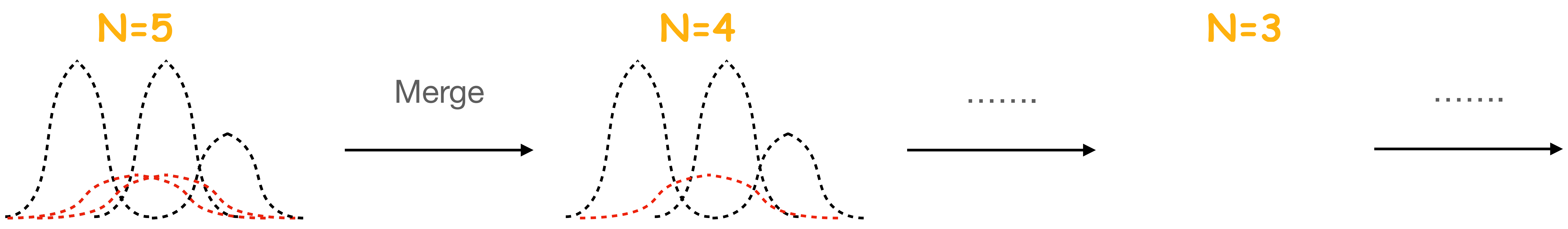
$$\tilde{G} = \text{argmin}_{G^\dagger \in \mathbb{G}_M} \int \{\phi(x;G) - \phi(x;G^\dagger)\}^2 dx$$

- **Clustering-based** (*Schieferdecker and Huber, 2009; Assa and Plataniotis, 2018*)



Existing approaches

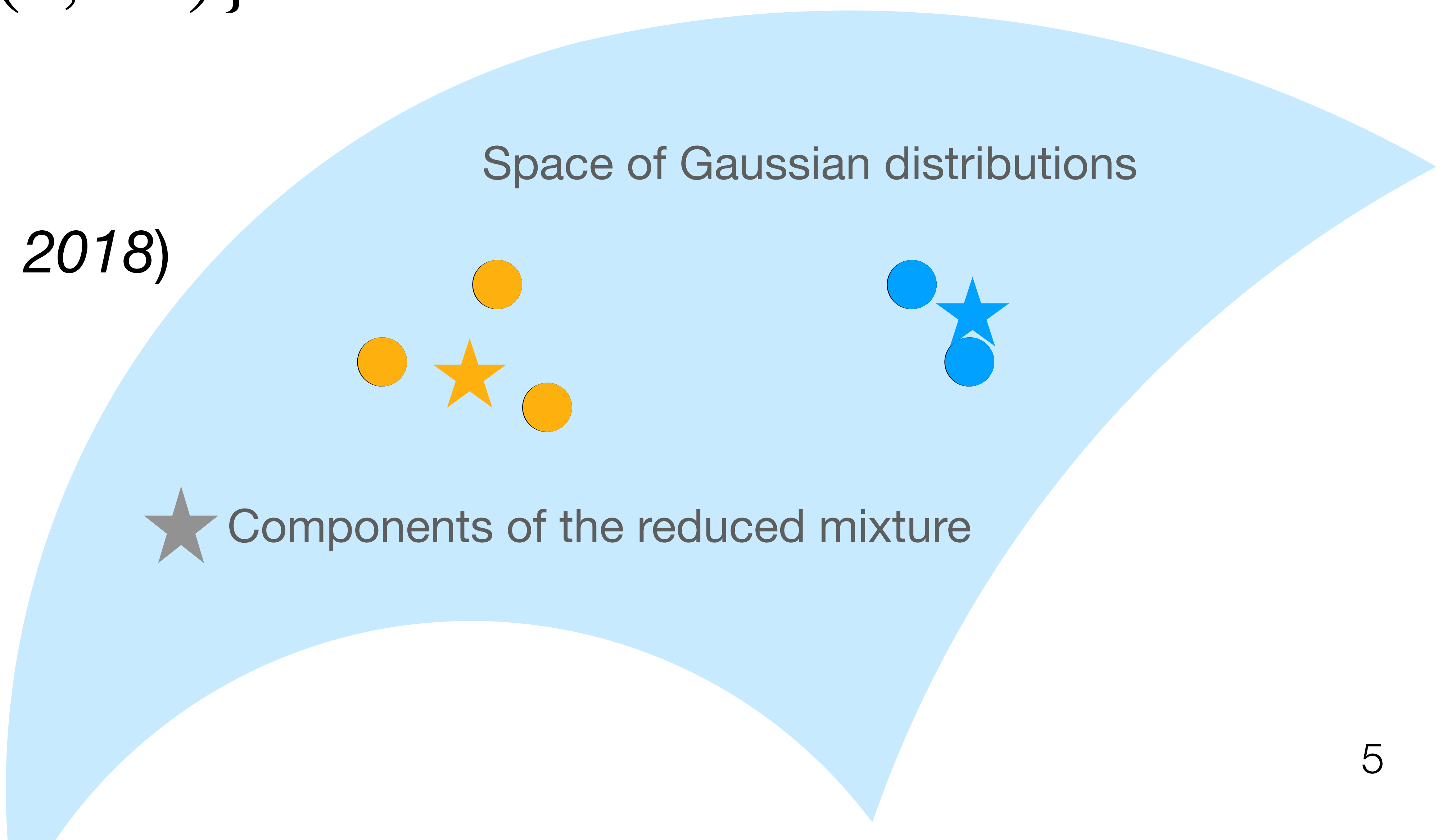
- **Greedy algorithm** (*Salmond, 1990; Runnalls, 2007; Assa and Plataniotis, 2018*)



- **Optimization-based** (*Williams and Maybeck, 2006*): directly search for

$$\tilde{G} = \text{argmin}_{G^\dagger \in \mathbb{G}_M} \int \{\phi(x; G) - \phi(x; G^\dagger)\}^2 dx$$

- **Clustering-based** (*Schieferdecker and Huber, 2009; Assa and Plataniotis, 2018*)



Existing approaches: pros & cons

Approach	Pros and cons
Greedy	✓Fast computation ✗Sub-optimal solution
Optimization-based	✓Clear optimality target ✗Heavy computation: $\mathcal{O}(NMd^3 + d^4)$ per iteration
Clustering-based	✓Fast computation: $\mathcal{O}(NMd^3)$ per iteration ✗Unclear optimality target ✗Unknown algorithm convergence

Existing approaches: pros & cons

Approach	Pros and cons
Greedy	✓Fast computation ✗Sub-optimal solution
Optimization-based	✓Clear optimality target ✗Heavy computation: $\mathcal{O}(NMd^3 + d^4)$ per iteration
Clustering-based	✓Fast computation: $\mathcal{O}(NMd^3)$ per iteration ✗Unclear optimality target Contribution 1: find a general optimization objective ✗Unknown algorithm convergence

Existing approaches: pros & cons

Approach	Pros and cons
Greedy	✓ Fast computation ✗ Sub-optimal solution
Optimization-based	✓ Clear optimality target ✗ Heavy computation: $\mathcal{O}(NMd^3 + d^4)$ per iteration
Clustering-based	✓ Fast computation: $\mathcal{O}(NMd^3)$ per iteration ✗ Unclear optimality target — Contribution 1: find a general optimization objective ✗ Unknown algorithm convergence — Contribution 2: establish algorithm convergence

Proposed method

Entropic regularized composite transportation divergence

- Let $c(\cdot, \cdot)$ be a divergence on the space of Gaussian distributions
- The entropic regularized composite transportation divergence between $\phi(x; G)$ and $\phi(x; \tilde{G})$ is defined to be

$$\mathcal{T}_c^{\lambda}(\phi(\cdot; G), \phi(\cdot; \tilde{G})) = \min \left\{ \sum_{n,m} \pi_{nm} c(\phi_n, \tilde{\phi}_m) - \lambda \mathcal{H}(\pi) : \sum_m \pi_{nm} = w_n, \sum_n \pi_{nm} = \tilde{w}_m \right\}$$

- A byproduct of the optimal transportation theory

Proposed method

Entropic regularized composite transportation divergence

- Let $c(\cdot, \cdot)$ be a divergence on the space of Gaussian distributions
- The entropic regularized composite transportation divergence between $\phi(x; G)$ and $\phi(x; \tilde{G})$ is defined to be

$$\mathcal{T}_c^{\lambda}(\phi(\cdot\,; G), \phi(\cdot\,; \tilde{G})) = \min \left\{ \sum_{n,m} \pi_{nm} c(\phi_n, \tilde{\phi}_m) - \lambda \boxed{\mathcal{H}(\pi)} : \sum_m \pi_{nm} = w_n, \sum_n \pi_{nm} = \tilde{w}_m \right\}$$

Entropy

- A byproduct of the optimal transportation theory

Proposed method

Entropic regularized composite transportation divergence

- Let $c(\cdot, \cdot)$ be a divergence on the space of Gaussian distributions
- The entropic regularized composite transportation divergence between $\phi(x; G)$ and $\phi(x; \tilde{G})$ is defined to be

$$\mathcal{T}_c^{\lambda}(\phi(\cdot\,; G), \phi(\cdot\,; \tilde{G})) = \min \left\{ \sum_{n,m} \pi_{nm} c(\phi_n, \tilde{\phi}_m) - \lambda \underbrace{\mathcal{H}(\pi)}_{\text{Entropy}} : \sum_m \pi_{nm} = w_n, \sum_n \pi_{nm} = \tilde{w}_m \right\}$$

- A byproduct of the optimal transportation theory

- Our proposed reduction mixture is

$$\tilde{G} = \operatorname{argmin}_{G^{\dagger} \in \mathbb{G}_M} \mathcal{T}_c^{\lambda}(\phi(\cdot\,; G), \phi(\cdot\,; G^{\dagger}))$$

- We proposed a class of methods for various choices of the divergence $c(\cdot, \cdot)$

Our MM algorithm

1. Assignment step

$$\pi_{nm}^{\lambda}(G^{(t)}) = w_n \frac{\exp(c(\phi_n, \phi_m^{(t)})/\lambda)}{\sum_k \exp(c(\phi_n, \phi_k^{(t)})/\lambda)}$$

2. Update step

$$\phi_m^{(t+1)} = \text{argmin}_{\phi} \sum_{n=1}^{N} \pi_{nm}^{\lambda}(G^{(t)}) c(\phi_n, \phi)$$

$$w_m^{(t+1)} = \sum_{n=1}^{N} \pi_{nm}^{\lambda}$$

Our MM algorithm

1. Assignment step

$$\boxed{\pi^{\lambda}_{nm}(G^{(t)})} = w_n \frac{\exp(c(\phi_n, \phi_m^{(t)})/\lambda)}{\sum_k \exp(c(\phi_n, \phi_k^{(t)})/\lambda)}$$

Assignment plan

2. Update step

$$\phi_m^{(t+1)} = \text{argmin}_{\phi} \sum_{n=1}^{N} \pi^{\lambda}_{nm}(G^{(t)}) c(\phi_n, \phi)$$

$$w_m^{(t+1)} = \sum_{n=1}^{N} \pi^{\lambda}_{nm}$$

Our MM algorithm

1. Assignment step

$$\pi^{\lambda}_{nm}(G^{(t)}) = w_n \frac{\exp(c(\phi_n, \phi^{(t)}_m)/\lambda)}{\sum_k \exp(c(\phi_n, \phi^{(t)}_k)/\lambda)}$$

Assignment plan

Hard clustering as $\lambda \to 0$

2. Update step

$$\phi^{(t+1)}_m = \text{argmin}_{\phi} \sum_{n=1}^{N} \pi^{\lambda}_{nm}(G^{(t)}) c(\phi_n, \phi)$$

$$w^{(t+1)}_m = \sum_{n=1}^{N} \pi^{\lambda}_{nm}$$

Our MM algorithm

1. Assignment step

$$\pi^{\lambda}_{nm}(G^{(t)}) = w_n \frac{\exp(c(\phi_n, \phi_m^{(t)})/\lambda)}{\sum_k \exp(c(\phi_n, \phi_k^{(t)})/\lambda)}$$

Assignment plan

Hard clustering as $\lambda \to 0$

2. Update step

$$\phi_m^{(t+1)} = \text{argmin}_{\phi} \sum_{n=1}^{N} \pi^{\lambda}_{nm}(G^{(t)}) c(\phi_n, \phi)$$

- Barycenter on space of Gaussian distributions
- Have closed-form solutions for certain choices of $c(\cdot, \cdot)$ such as the KL divergence

$$w_m^{(t+1)} = \sum_{n=1}^{N} \pi^{\lambda}_{nm}$$

Algorithm convergence

- For hard clustering ($\lambda = 0$), worst case M^N iterations in theory and only 2-3 iterations in practice
- For soft clustering ($\lambda > 0$), analysis using mirror descent
- The MM update can be written as

$$G^{(t+1)} = \text{argmin}_G \left\{ \mathcal{J}_c^\lambda(G^{(t)}) + \langle \nabla \mathcal{J}_c^\lambda(G^{(t)}), G - G^{(t)} \rangle + \sum_{m=1}^{M} \pi_{\cdot m}^\lambda(G^{(t)}) D_A(\theta_m, \theta_m^{(t)}) \right\}$$

- Linear convergence

$$\min_{t \le T} \sum_{n,m} \pi_{nm}^\lambda(G^{(t)}) D_A(\theta_m^{(t)}, \theta_m^{(t+1)}) \le \frac{\mathcal{J}_c^\lambda(G^{(0)}) - \mathcal{J}_c^*}{T}$$

Real data-hand gesture recognition

10 comp mixture

Real data-hand gesture recognition

Build class prototype

10 comp mixture

Real data-hand gesture recognition

10 comp mixture

Build class prototype

Real data-hand gesture recognition

10 comp mixture

Build class prototype

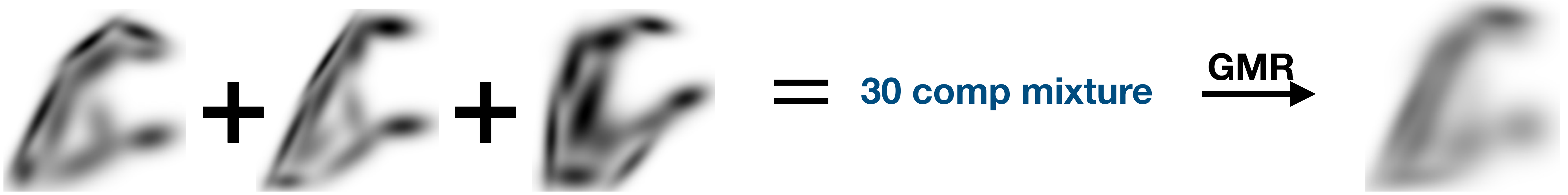
= 30 comp mixture

Real data-hand gesture recognition

10 comp mixture

Build class prototype

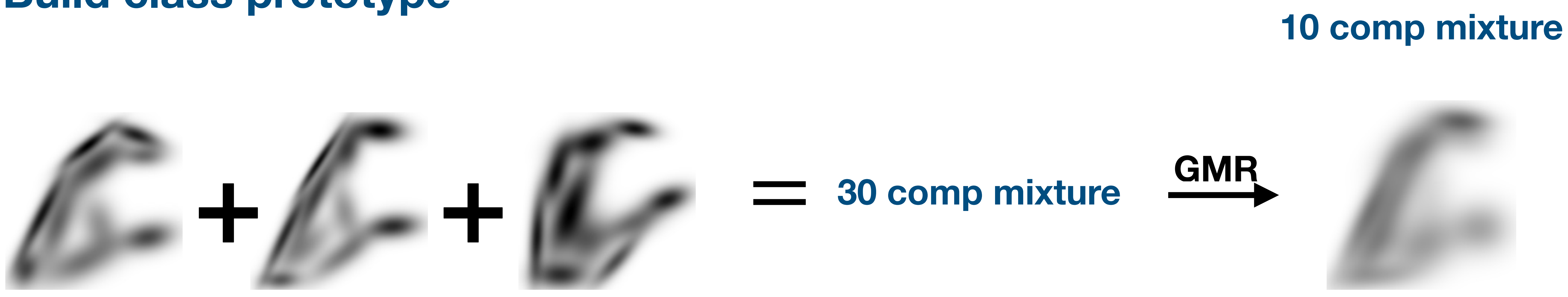
10 comp mixture



Real data-hand gesture recognition

10 comp mixture

Build class prototype

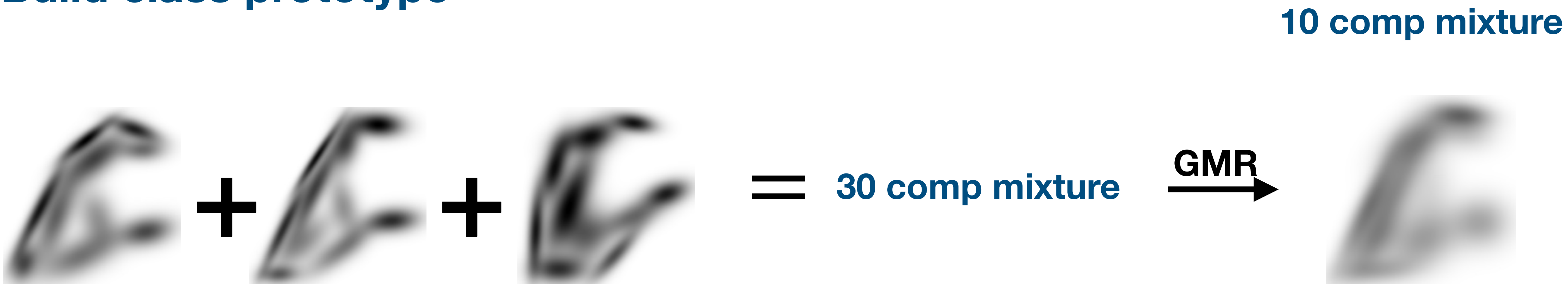


Classify new images (closest divergence to prototype)

Real data-hand gesture recognition

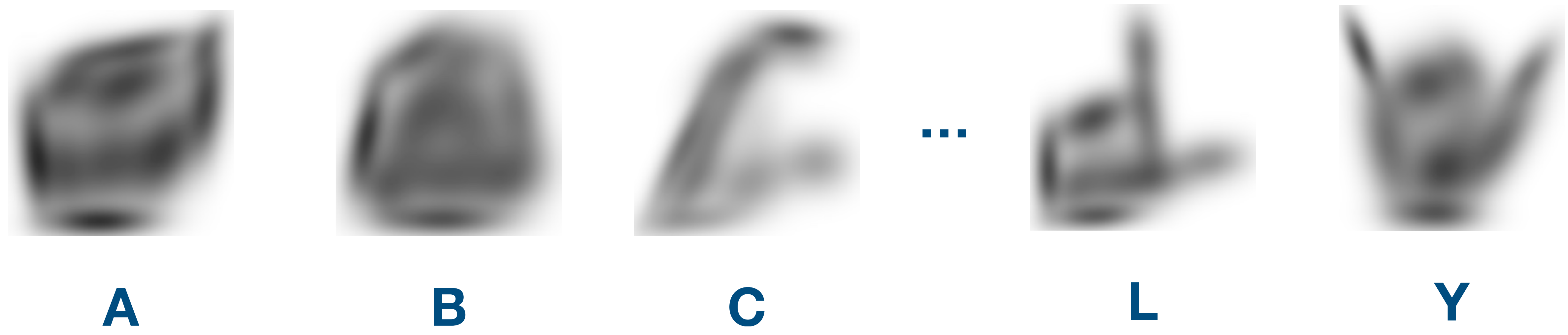
10 comp mixture

Build class prototype



Classify new images (closest divergence to prototype)

Prototype (Only 10 images)

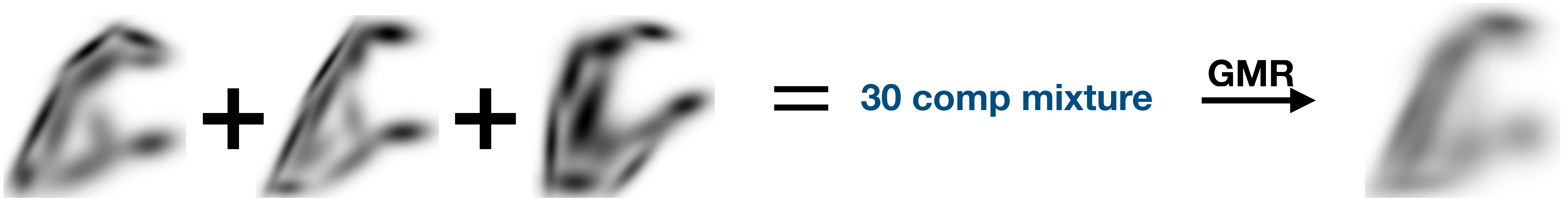


Real data-hand gesture recognition

10 comp mixture

Build class prototype

10 comp mixture



Classify new images (closest divergence to prototype)

Prototype (Only 10 images)

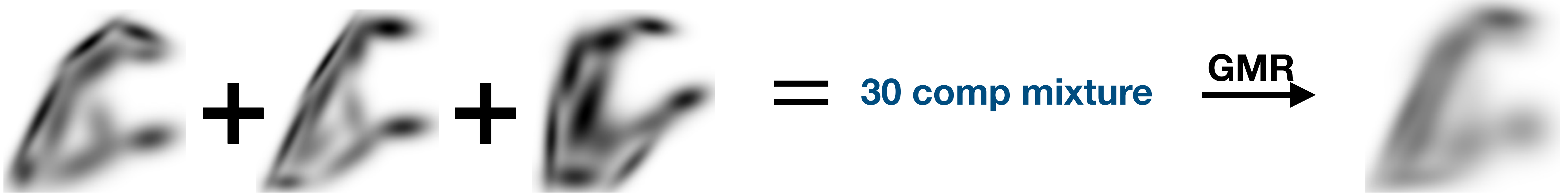
A B C ... L Y

Test image

Real data-hand gesture recognition

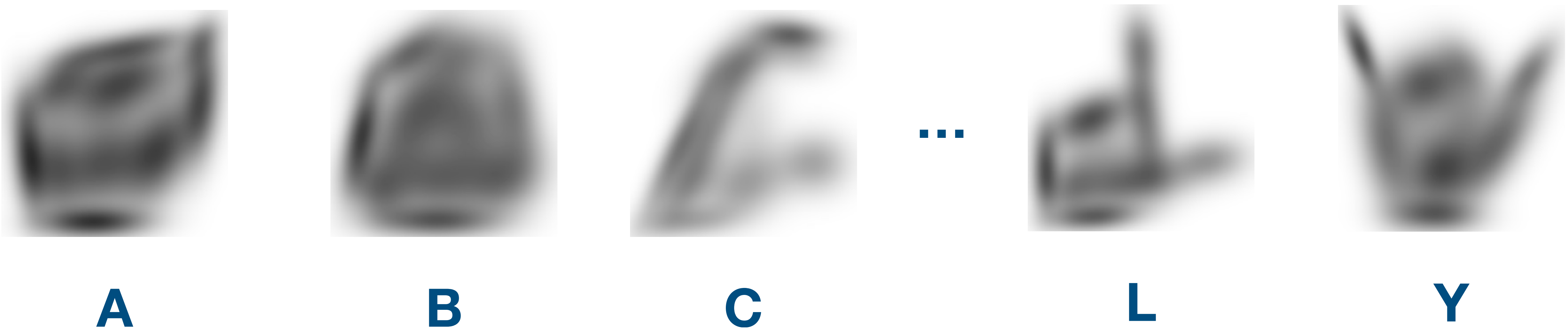
10 comp mixture

Build class prototype



Classify new images (closest divergence to prototype)

Prototype (Only 10 images)



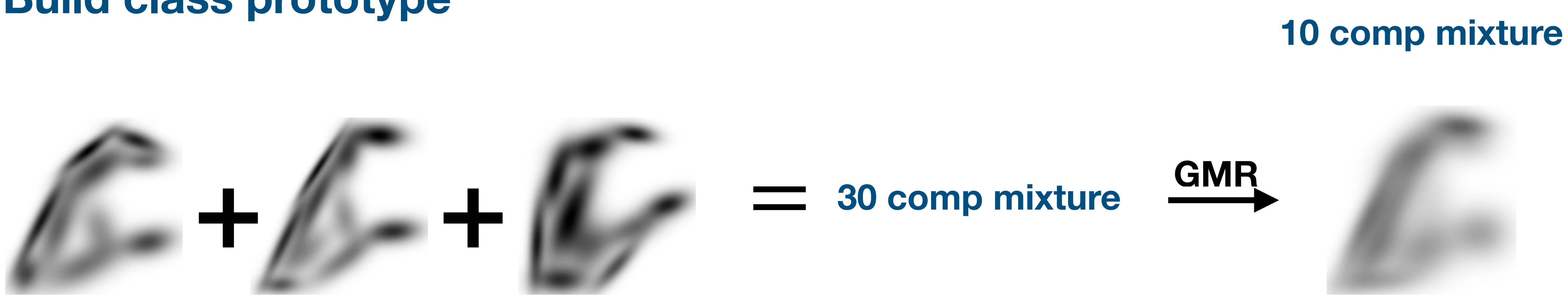
Test image

This is an “L”!

Real data-hand gesture recognition

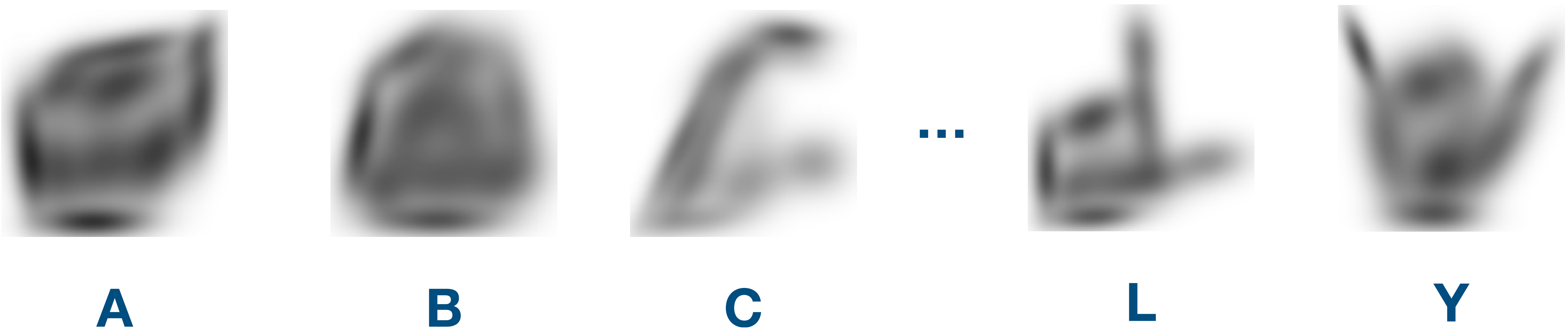
10 comp mixture

Build class prototype



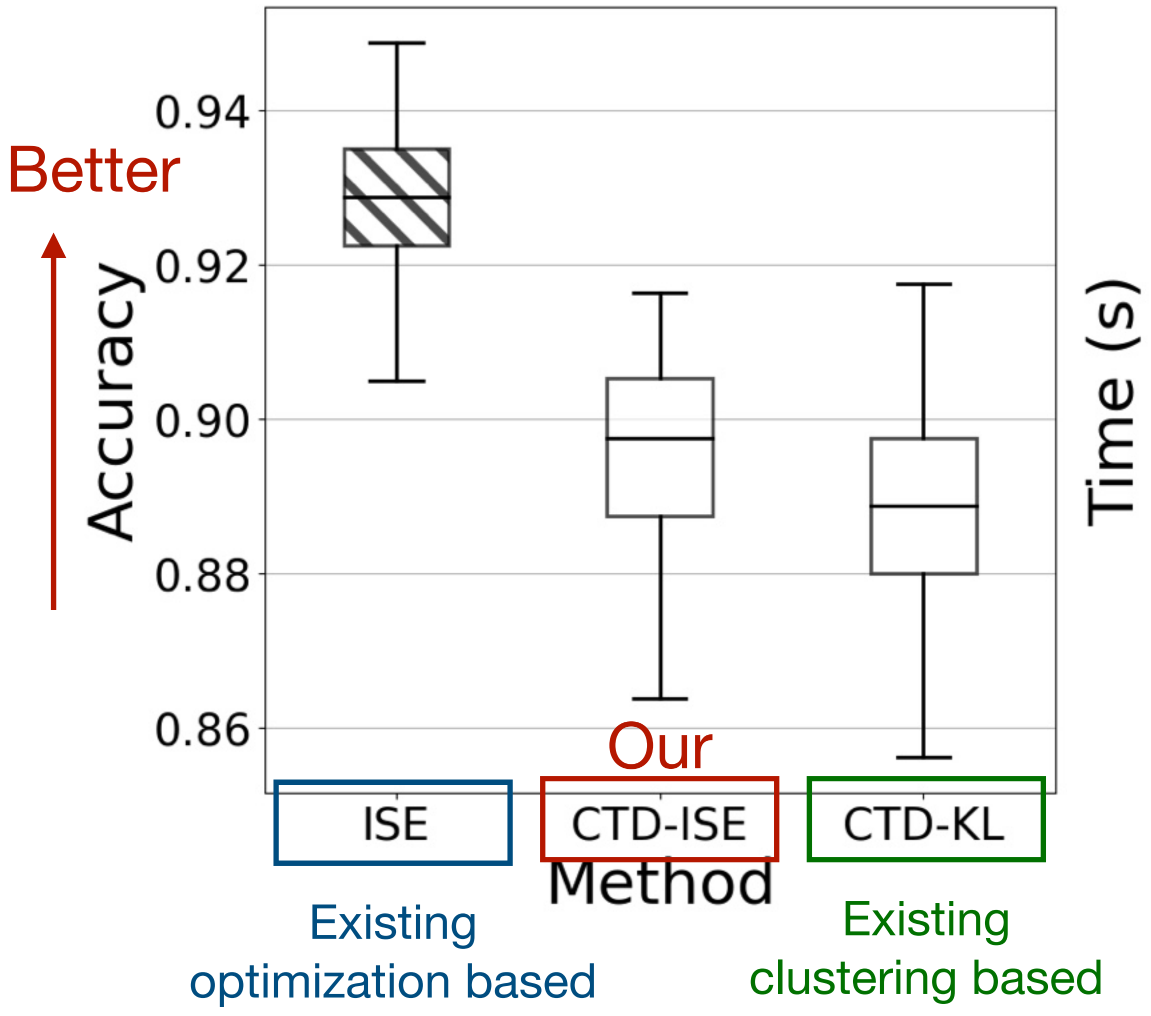
Classify new images (closest divergence to prototype)

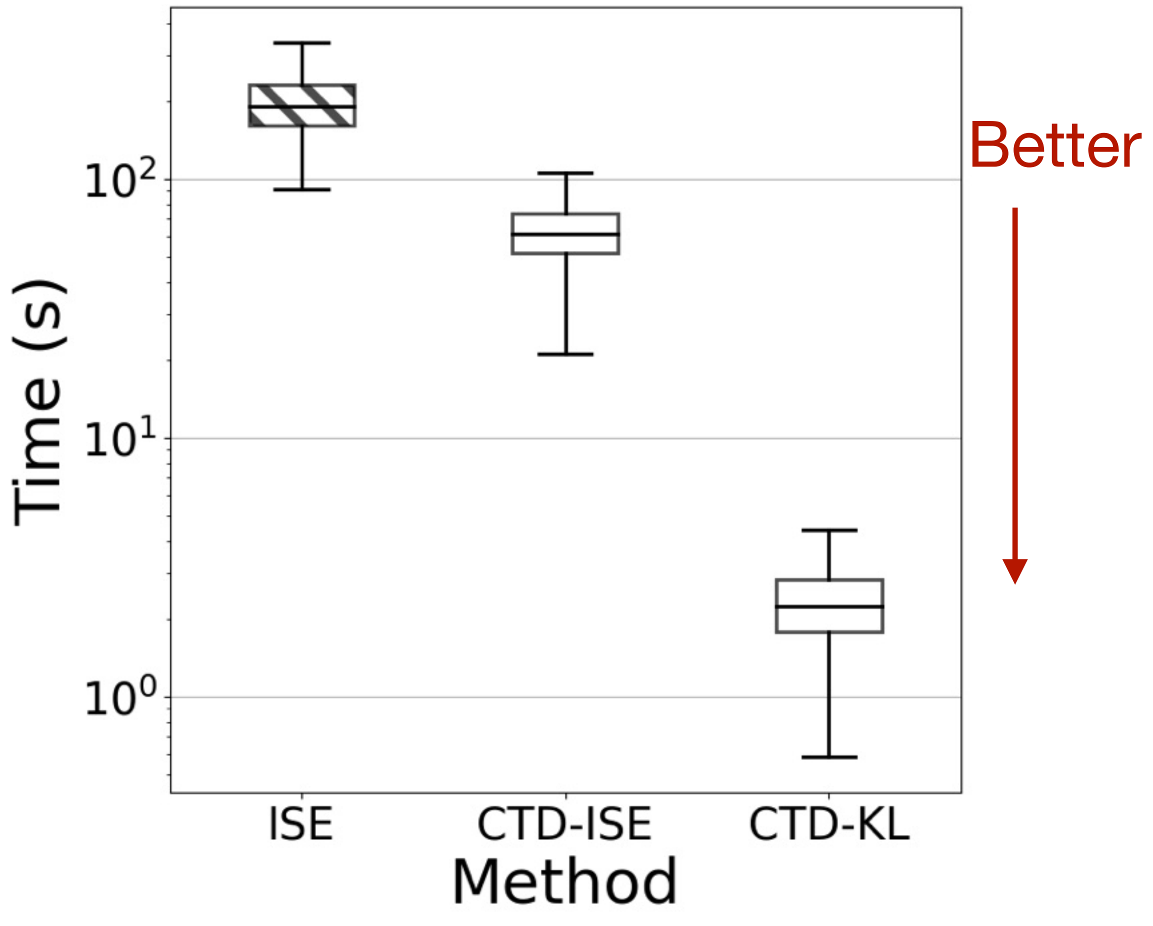
Prototype (Only 10 images)



Test image

This is an “L”!





Summary of our contribution

- We connect the existing clustering algorithms with the MM algorithm
- Establish the theoretical guarantees for the existing approach
- Reduction performance: the ISE is the optimal cost function among several choices