Multimodal nonlinear filtering using Gauss-Hermite Quadrature

Hannes Saal, Nicolas Heess, Sethu Vijayakumar

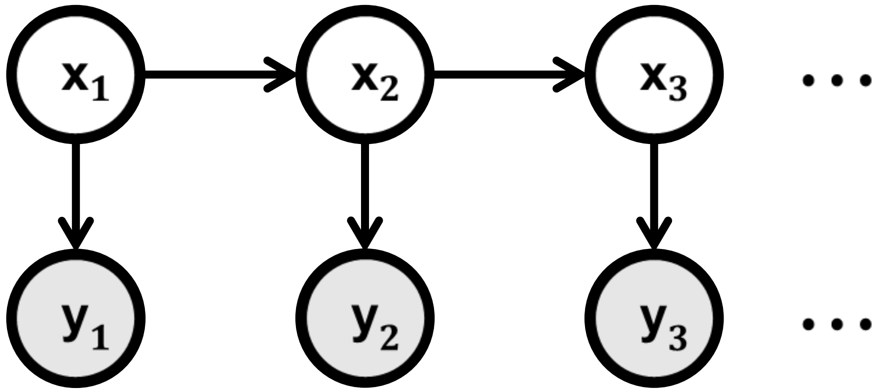
University of Edinburgh

THE UNIVERSITY OF EDINBURGH

ianc | Institute for Adaptive and Neural Computation

ipab | Institute of Perception, Action and Behaviour

The filtering problem



Given a sequence of observ $p(x_t|y_t, y_{t-1} \dots y_1)$

Recursive formulation:

$$p(x_t|y_{t-1} \dots y_1) = \int p(x_t|x_{t-1}) p(x_{t-1}|y_{t-1} \dots y_1) \mathrm{d}\, x_{t-1}$$ (Time update)

$$p(x_t|y_t, y_{t-1} \dots y_1) = \frac{1}{Z_t} p(y_t|x_t) p(x_t|y_{t-1} \dots y_1)$$

$$Z_t = \int p(y_t|x) p(x_t|y_{t-1} \dots y_1) \mathrm{d}x$$

(Observation update)

Filtering with nonlinear likelihood

Simplified scenario without state dynamics (observation update only):

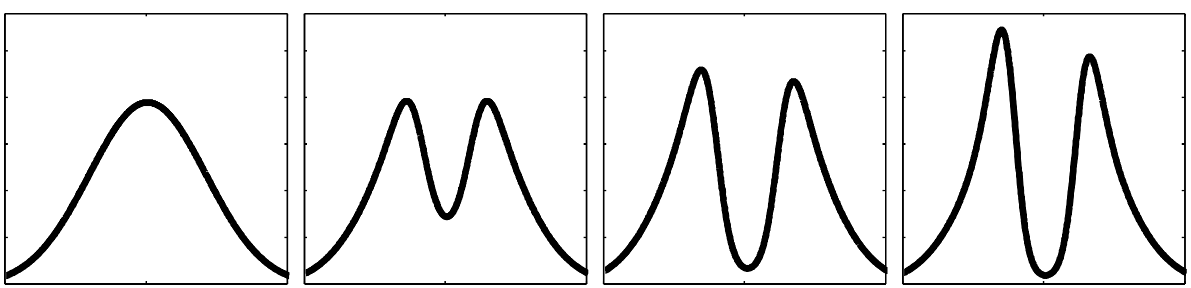
$$p(x|y_t, y_{t-1} \dots y_1) = \frac{1}{Z_t} p(y_t|x) p(x|y_{t-1} \dots y_1)$$

With a non-linear likelihood, e.g. $p(y|x) = N(y|f(x), \Sigma)$ for some nonlinear $f(x)$, the posterior can have a complex shape and be multimodal

1. **How to approximately represent** $q_t(x) \approx p(x|y_t, y_{t-1} \dots y_1)$ **?**

2. **Given** $q_{t-1}(x)$, y_t **how to update the approximate representation?**

$$q_t(x) \approx \frac{1}{Z_t} p(y_t|x) q_{t-1}(x)$$



$p(x)$ $\quad$ $p(x|y_1)$ $\quad$ $p(x|y_2, y_1)$ $\quad$ $p(x|y_3, y_2, y_1)$

Approximate representations of the posterior

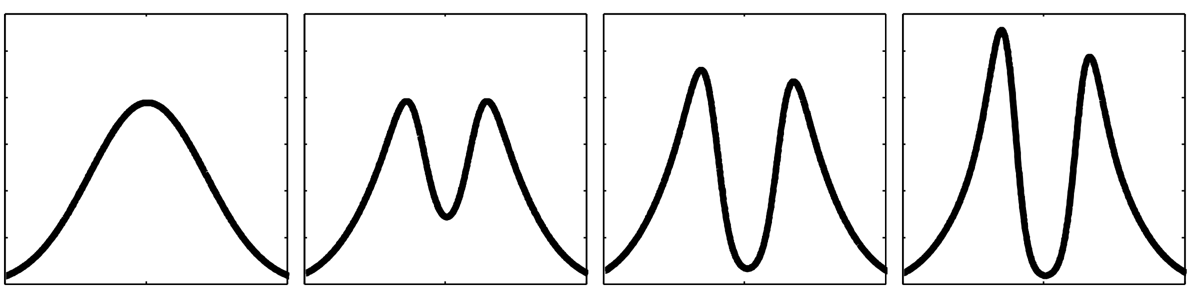
Sample-based representations:

- Particle Filtering: $q_t(x) = \sum_{l=1}^{L} w_l \delta(x - \hat{x}_l)$

Deterministic approximations:

- ***Gaussian*** distribution: $q_t(x) = \mathrm{N}(\mathrm{x}|\mu, \Sigma)$ (e.g. Extended Kalman Filter)
  recursive update: linearization of the observation function

- ***Mixture of Gaussians***: $q_t(x) = \sum_m \alpha_m N(x|\mu_m, \Sigma_m)$ (e.g. Gaussian Sum Filter)
  update: parameters of mixture components updated *independently*
  ("Bank of independent filters")



$p(x)$ $p(x|y_1)$ $p(x|y_2, y_1)$ $p(x|y_3, y_2, y_1)$

Approximate representations of the posterior

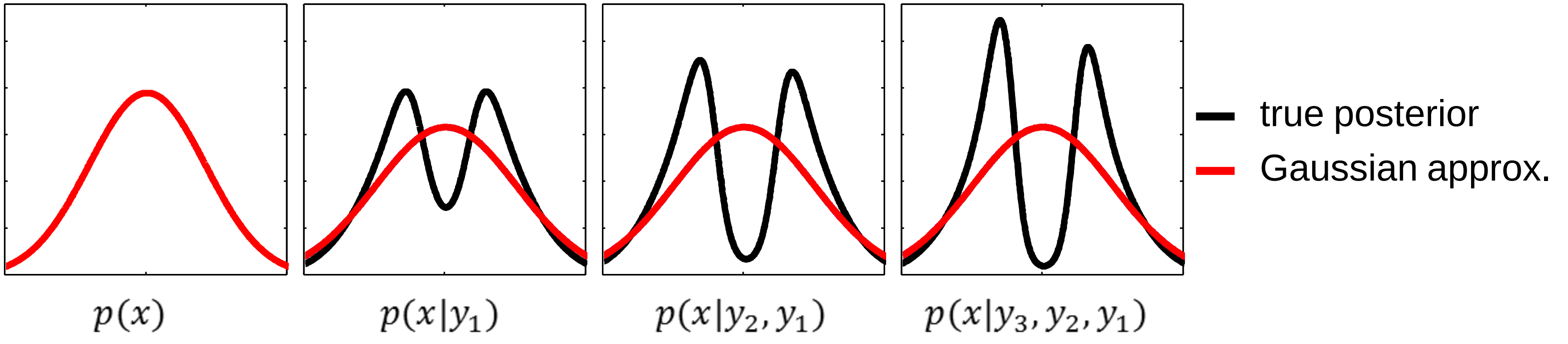
Sample-based representations:

- Particle Filtering: $q_t(x) = \sum_{l=1}^{L} w_l \delta(x - \hat{x}_l)$

Deterministic approximations:

- ***Gaussian*** distribution: $q_t(x) = \mathrm{N}(\mathrm{x}|\mu, \Sigma)$ (e.g. Extended Kalman Filter)
  recursive update: linearization of the observation function

- ***Mixture of Gaussians***: $q_t(x) = \sum_m \alpha_m N(x|\mu_m, \Sigma_m)$ (e.g. Gaussian Sum Filter)
  update: parameters of mixture components updated *independently*
  ("Bank of independent filters")



Approximate representations of the posterior

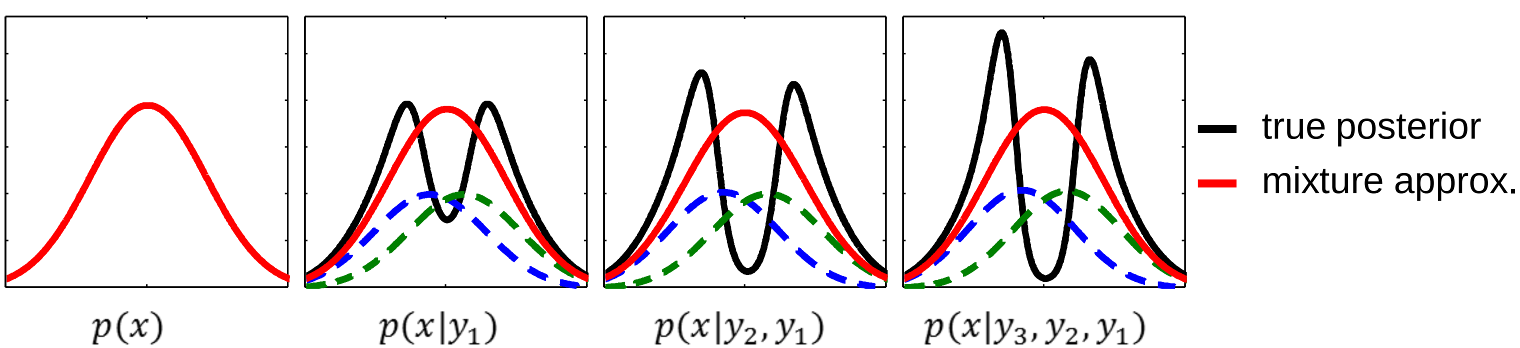
Sample-based representations:

- Particle Filtering: $q_t(x) = \sum_{l=1}^{L} w_l \delta(x - \hat{x}_l)$

Deterministic approximations:

- ***Gaussian*** distribution: $q_t(x) = \mathrm{N}(\mathrm{x}|\mu, \Sigma)$ (e.g. Extended Kalman Filter)
  recursive update: linearization of the observation function

- ***Mixture of Gaussians***: $q_t(x) = \sum_m \alpha_m N(x|\mu_m, \Sigma_m)$ (e.g. Gaussian Sum Filter)
  update: parameters of mixture components updated *independently*
  ("Bank of independent filters")



Approximate representations of the posterior

Sample-based representations:

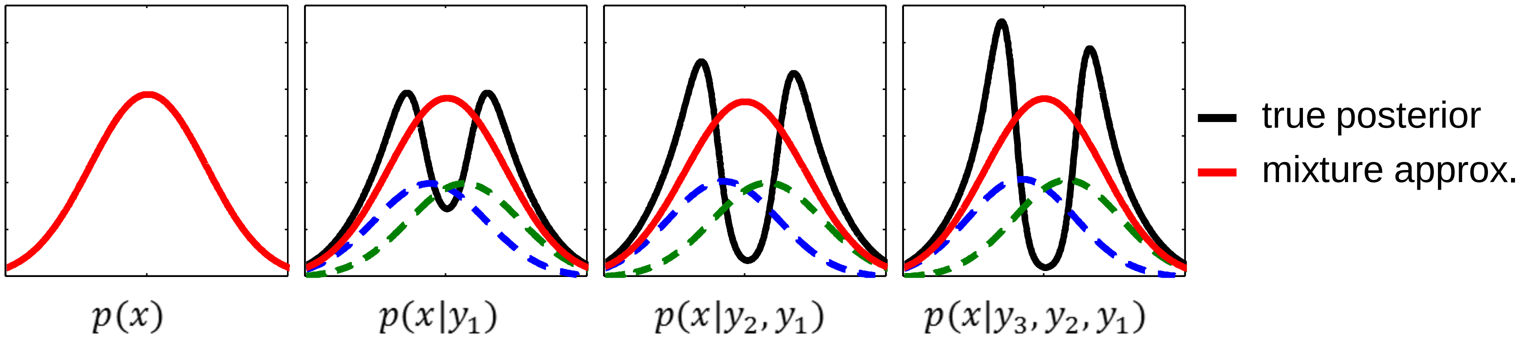
- Particle Filtering: $q_t(x) = \sum_{l=1}^{L} w_l \delta(x - \hat{x}_l)$

Deterministic approximations:

- ***Gaussian*** distribution: $q_t(x) = \mathrm{N}(\mathrm{x}|\mu, \Sigma)$ (e.g. Extended Kalman Filter)
  recursive update: linearization of the observation function

- ***Mixture of Gaussians***: $q_t(x) = \sum_m \alpha_m N(x|\mu_m, \Sigma_m)$ (e.g. Gaussian Sum Filter)
  update: parameters of mixture components updated *independently*
  ("Bank of independent filters")

- **Variational mixture filter:** Mixture of Gaussians
  update: joint optimization of all mixture parameters



Approximate representations of the posterior

Sample-based representations:

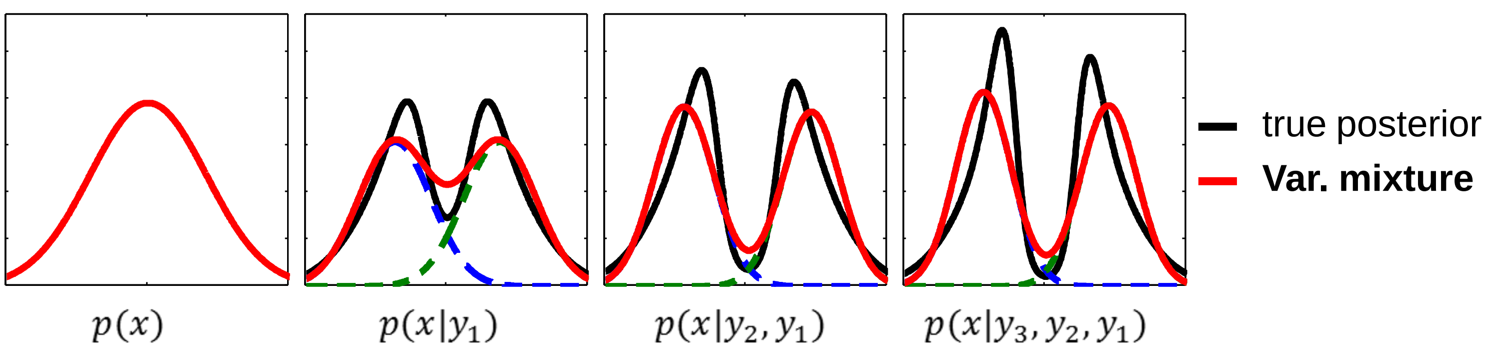
- Particle Filtering: $q_t(x) = \sum_{l=1}^{L} w_l \delta(x - \hat{x}_l)$

Deterministic approximations:

- ***Gaussian*** distribution: $q_t(x) = \mathrm{N}(\mathrm{x}|\mu, \Sigma)$ (e.g. Extended Kalman Filter)
  recursive update: linearization of the observation function

- ***Mixture of Gaussians***: $q_t(x) = \sum_m \alpha_m N(x|\mu_m, \Sigma_m)$ (e.g. Gaussian Sum Filter)
  update: parameters of mixture components updated *independently*
  ("Bank of independent filters")

- **Variational mixture filter:** Mixture of Gaussians
  update: joint optimization of all mixture parameters



Variational mixture filter

An efficient way of using Mixtures of Gaussians to compactly represent complex state posterior distributions

- **State prior**: $p(x) = \sum_n \gamma_n N(x|\nu_n, \Theta_n)$
- **Nonlinear likelihood**: $p(y|x) = N(y|f(x), \Sigma)$
- Gaussian mixture representation of the **approximate posterior**:
  $q_t(x) = \sum_m \alpha_m N(x|\mu_m, \Sigma_m)$

- Joint optimization of the mixture parameters through ***minimization of the Kullback-Leibler divergence***: $q_t(x) = \text{argmin}_q \, KL[\, q \,||\, \frac{1}{Z_t} p(y_t|x) q_{t-1}(x)]$

- Computational tractability:
  - ***Radial-Basis-Function representation*** of the observation function:
    $f(x) = \sum_j c_j k(x, m_j)$ where $k(x,m) = \exp\left\{-\frac{1}{2}(x-m)^T S^{-1}(x-m)\right\}$
  - ***Deterministic sampling*** for approximating intractable integrals

Optimizing the mixture representation

KL-divergence between the approximating mixture $q_t(x)$ and the exact recursive update $p(x|y_t, \ldots, y_1) \propto p(y_t|x)q_{t-1}(x)$

$$KL[q_t||p] = \int \mathrm{d}x\, q_t(x) \log \frac{q_t(x)}{p(x|y)}$$

$$= -\sum_m \alpha_m E_{N_m}[\log q_t(x)] - \sum_m \alpha_m E_{N_m}[\log q_{t-1}(x)\,]$$

$$-\sum_m \alpha_m E_{N_m}[\log p(y_t|x)] + \text{const}$$

Optimizing the mixture representation

KL-divergence between the approximating mixture $q_t(x)$ and the exact recursive update $p(x|y_t,\ldots,y_1) \propto p(y_t|x)q_{t-1}(x)$

$$KL[q_t||p] = \int \mathrm{d}x\, q_t(x) \log \frac{q_t(x)}{p(x|y)}$$

$$= -\sum_m \alpha_m E_{N_m}[\log q_t(x)] - \sum_m \alpha_m E_{N_m}[\log q_{t-1}(x)]$$

$$-\sum_m \alpha_m E_{N_m}[\log p(y_t|x)] + \text{const}$$

Intractable integrals!

Gauss-Hermite Quadrature

- Uses deterministic sampling to approximate integrals

$$\int N(x \,|\mu_m, \Sigma_m)\; g(x)\, \mathrm{d}x = E_{N_m}[g(x)] \;\approx \pi^{-\frac{d}{2}} \sum_h w_h g(z_h)$$

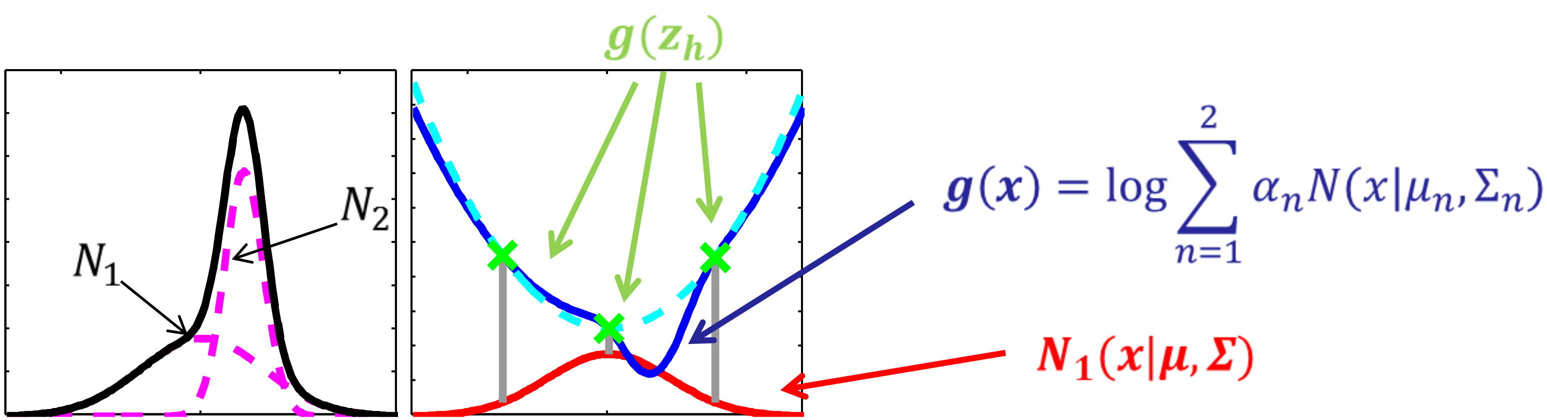
Gauss-Hermite Quadrature

- Uses deterministic sampling to approximate integrals

$$\int N(x\,|\mu_m, \Sigma_m)\; g(x)\, \mathrm{d}x = E_{N_m}[g(x)] \;\approx \pi^{-\frac{d}{2}} \sum_h w_h g(z_h)$$

- We are interested in expectations of logarithms of sums of Gaussians ("log-sums")

$$E_{N_m}[g(x)] = \; E_{N_m}\left[\log \sum_n \alpha_n N(x|\mu_n, \Sigma_n)\right]$$



Gaussian mixture   Approximation w/ GH

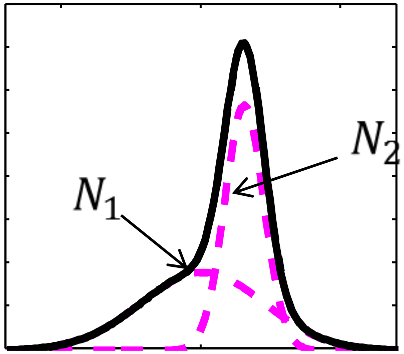
Gauss-Hermite Quadrature

- The approximation of log-sum expectation can be improved by decomposing the log-sum into a sum

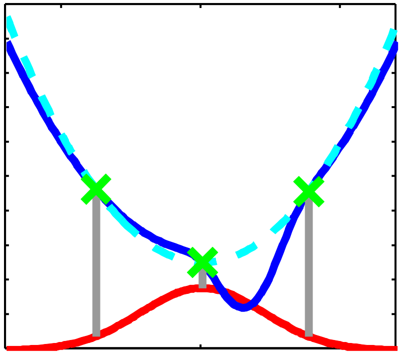
$$\log(N_1 + N_2 + N_3 + \cdots) = \log(N_1) + \log\left(1 + \frac{N_2}{N_1}\right) + \log\left(1 + \frac{\mathrm{N}_3}{\mathrm{N}_1 + \mathrm{N}_2}\right) + \cdots$$

- we can then optimize the approximation for each log-term individually

$$E_{N_1}\left[\log\left(1 + \frac{N_2}{N_1}\right)\right] = E_{N_2}\left[\frac{N_1}{N_2}\log\left(1 + \frac{N_2}{N_1}\right)\right]$$



Gaussian mixture

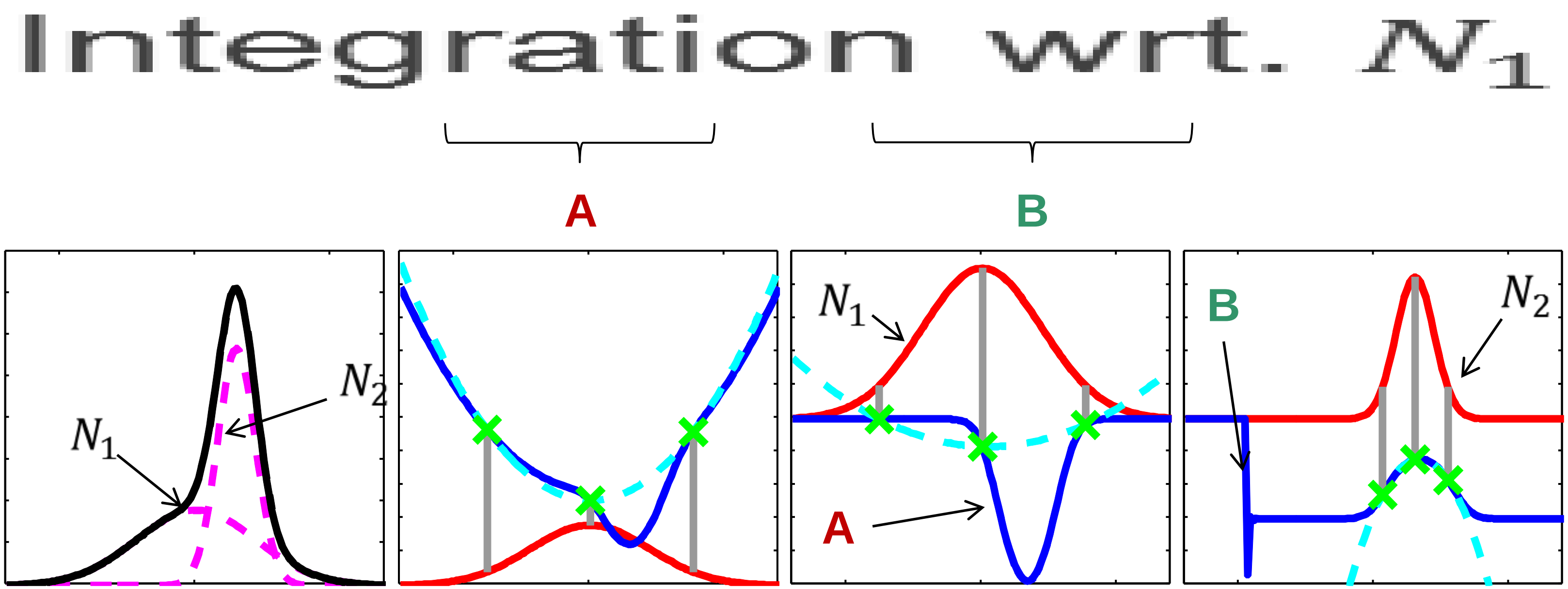

Direct approximation

Gauss-Hermite Quadrature

- The approximation of log-sum expectation can be improved by decomposing the log-sum into a sum

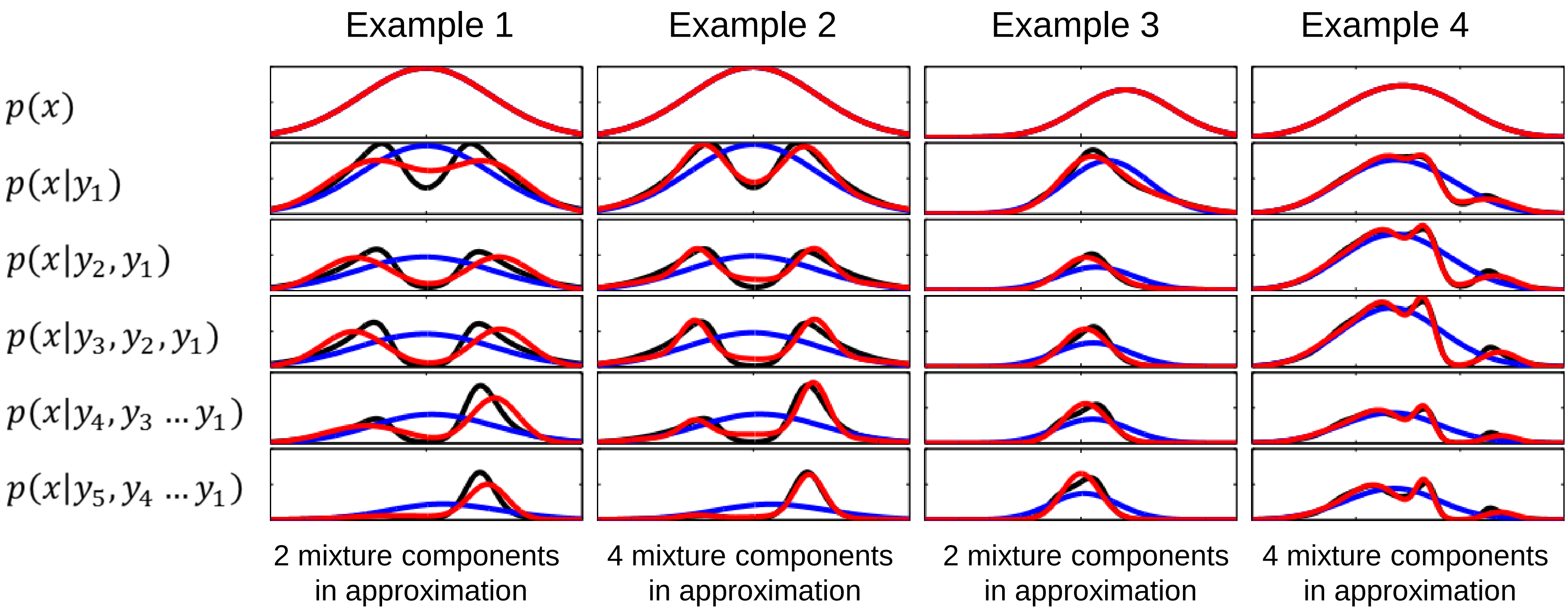
$$\log(N_1 + N_2 + N_3 + \cdots) = \log(N_1) + \log\left(1 + \frac{N_2}{N_1}\right) + \log\left(1 + \frac{N_3}{N_1 + N_2}\right) + \cdots$$

- we can then optimize the approximation for each log-term individually



Gaussian mixture | Direct approximation | Integration wrt. N_1 | Integration wrt. N_2

Qualitative Results: 1D



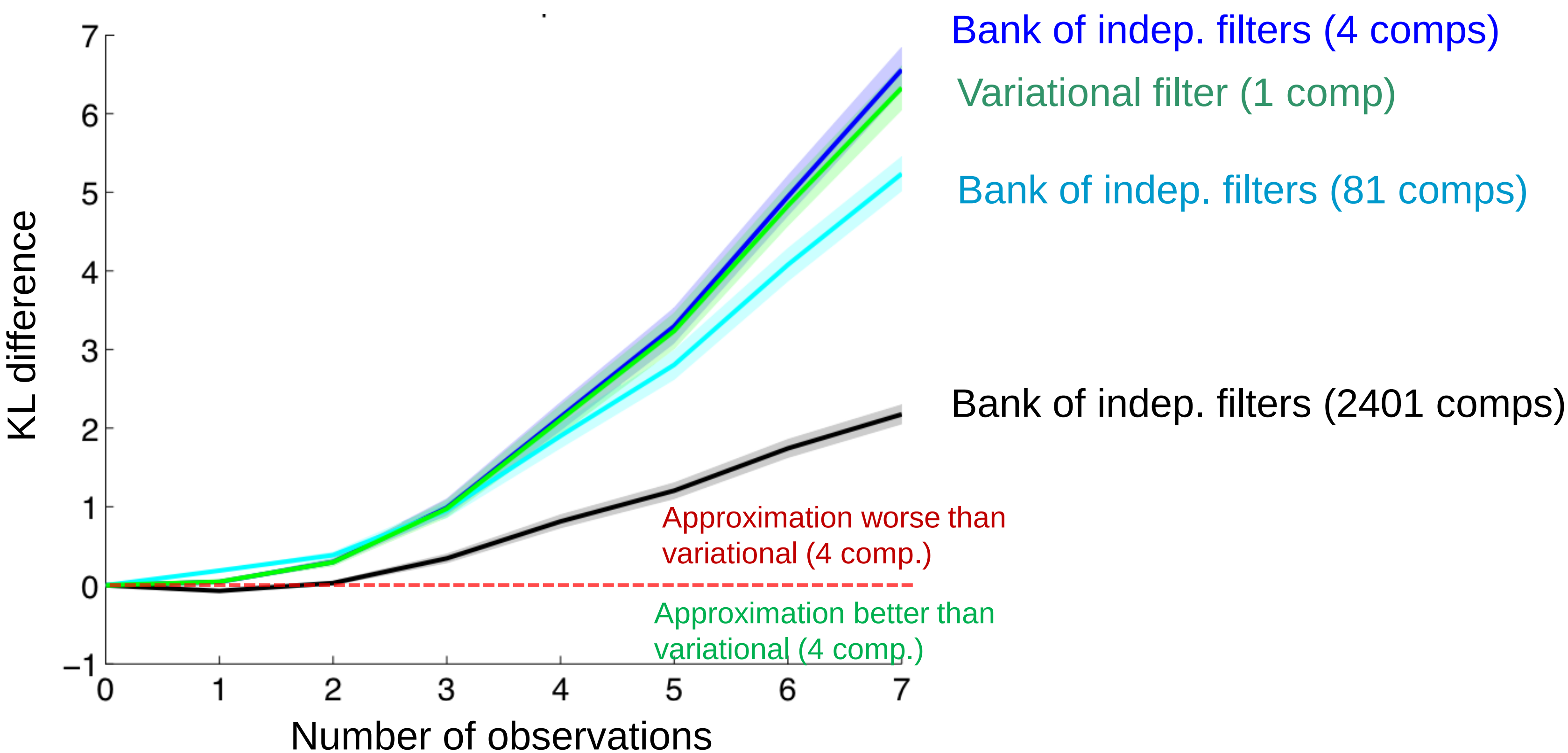
“True” posterior (computed numerically)

Mixture of Gaussians: variational approximation (our approach)

Mixture of Gaussians: bank of independent filters

Quantitative evaluation of approximation quality

Differences in KL divergence $KL[q_{alt}||p] - KL[q_{var}||p]$ between *alternative approaches* and *variational filter (4 mixture components)*

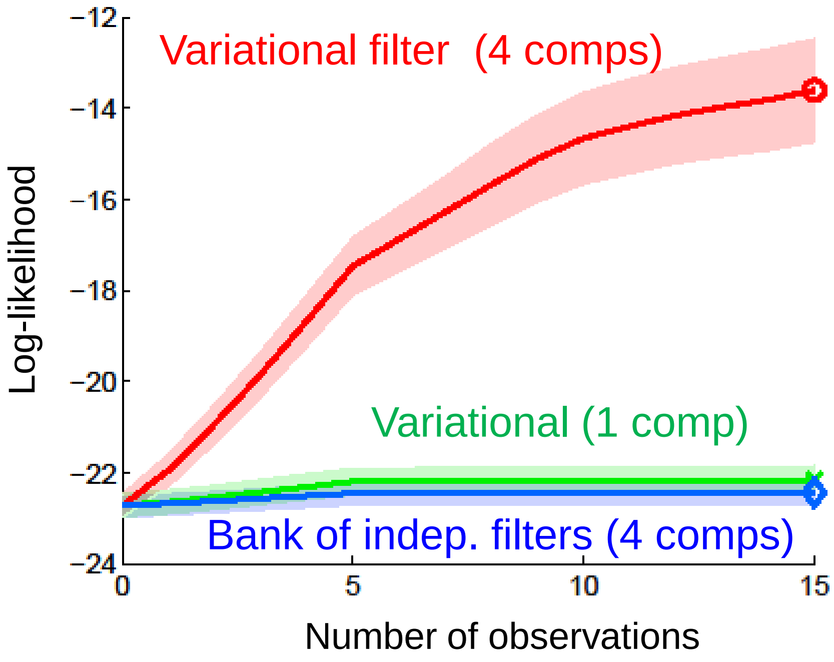


(state space with 4 dimensions; observation function with 3 RBF kernels; average over 100 runs with different settings of the model parameters)

Evaluation in the context of active learning

Localization task: static target x^*; observation function dependent on additional parameter controlling ***probe position*** $y = f(\theta - x^*)$

Convergence towards target:
Log-likelihood of target x^* under posterior

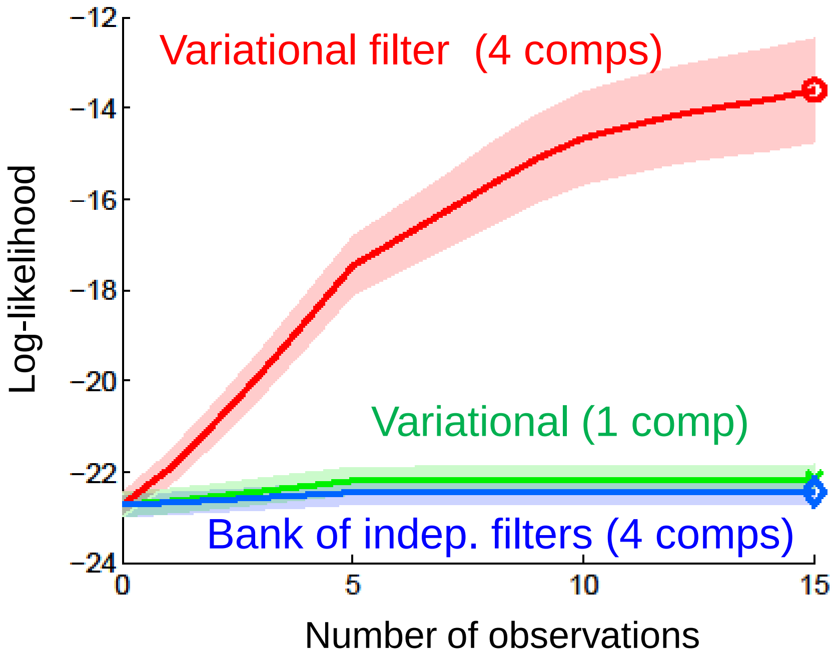


(state space: 10 dimensions; average over 25 runs with different parameters)

Evaluation in the context of active learning

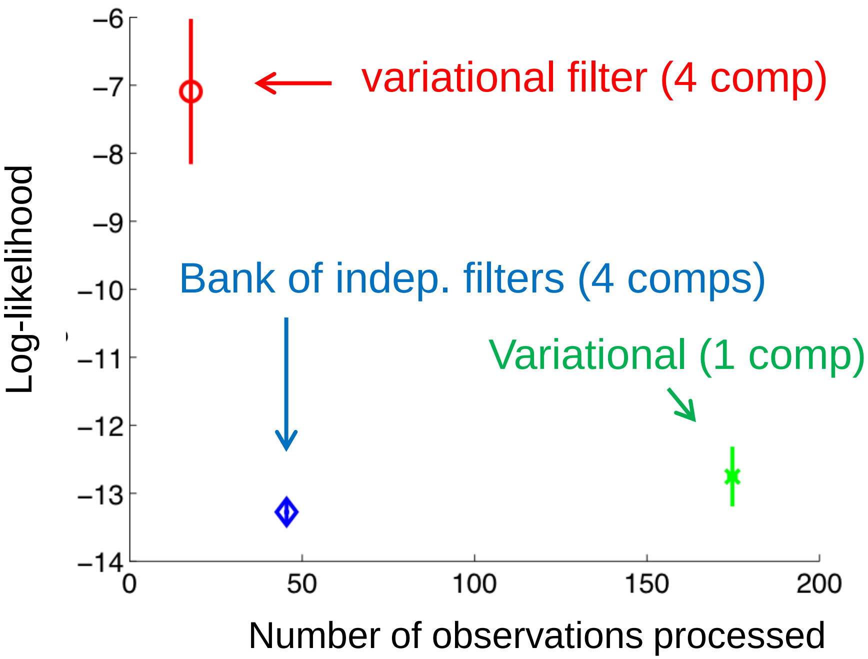
Localization task: static target x^*; observation function dependent on additional parameter controlling ***probe position*** $y = f(\theta - x^*)$

Convergence towards target:
Log-likelihood of target x^* under posterior



(state space: 10 dimensions; average over 25 runs with different parameters)

Time-accuracy tradeoff: fixed time budget for inference and active learning



(state space: 6 dimensions; 60 seconds available for inference; average over 25 runs)

Conclusions

- New approach to nonlinear filtering that efficiently fits mixture of Gaussians to posterior
- Compact parametric representation of the posterior
- Gradient based optimization of the KL divergence using Gauss-Hermite quadrature for efficient numerical integration
- Better fit to posterior as well as faster convergence
- Higher computational cost offset by better quality of approximation, especially important in active learning problems
- Some directions for future work:
  - Alternative observation nonlinearities (e.g. Gaussian processes)
  - Automatic selection of the # of mixture components

Thanks!

For questions:
hannes.saal@gmail.com

Gauss-Hermite Quadrature

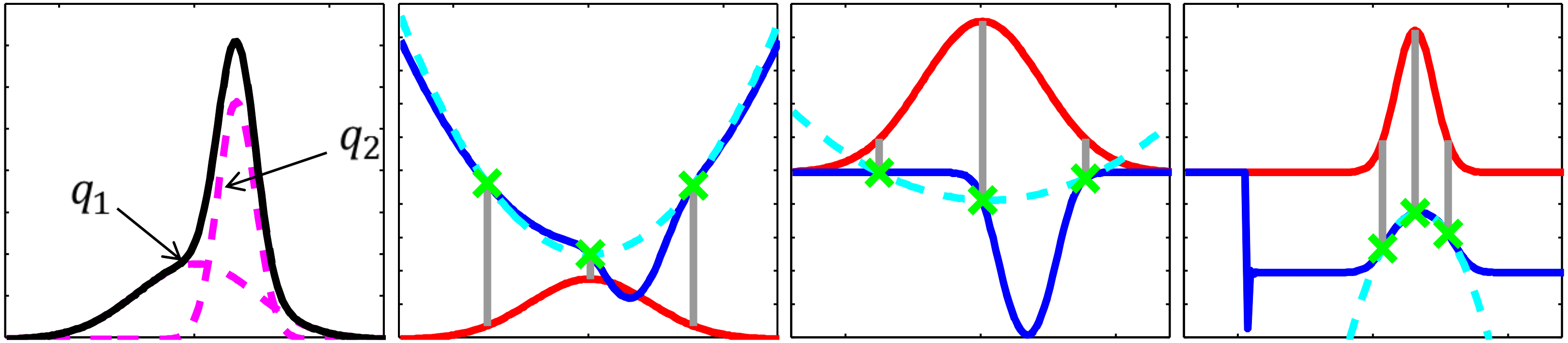
- The approximation of log-sum expectation can be improved by decomposing the log-sum into a sum

$$\log(N_1 + N_2 + N_3) = \log(N_1) + \log\left(1 + \frac{N_2}{N_1}\right) + \log\left(1 + \frac{\mathrm{N}_3}{\mathrm{N}_1 + \mathrm{N}_2}\right)$$

$$E_{q_m}\left[\log\left(\sum\nolimits_n \alpha_n N_n\right)\right] = E_{q_m}[\alpha_1 N_1] + \sum\nolimits_{i=2}^{N} E_{q_m}\left[\log\left(1 + \frac{\alpha_i N_i}{\sum_{k=1}^{i-1} \alpha_k N_k}\right)\right]$$

- Then we approximate individual components separately

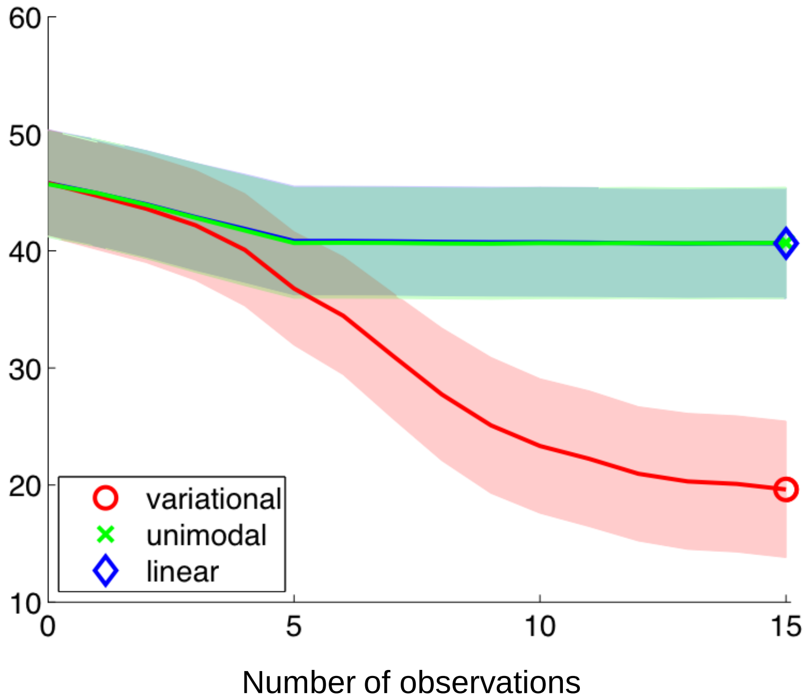
$$\underbrace{\mathrm{E}_{q_m}\left[\log\left(1 + \frac{\alpha_n \mathcal{N}_n}{\sum_{k=1}^{n-1} \alpha_k \mathcal{N}_k}\right)\right]}_{\text{Integral over } q_m} = \underbrace{\mathrm{E}_{q_n}\left[\frac{\mathcal{N}_m}{\mathcal{N}_n} \log\left(1 + \frac{\alpha_n \mathcal{N}_n}{\sum_{k=1}^{n-1} \alpha_k \mathcal{N}_k}\right)\right]}_{\text{Integral over } q_n}$$



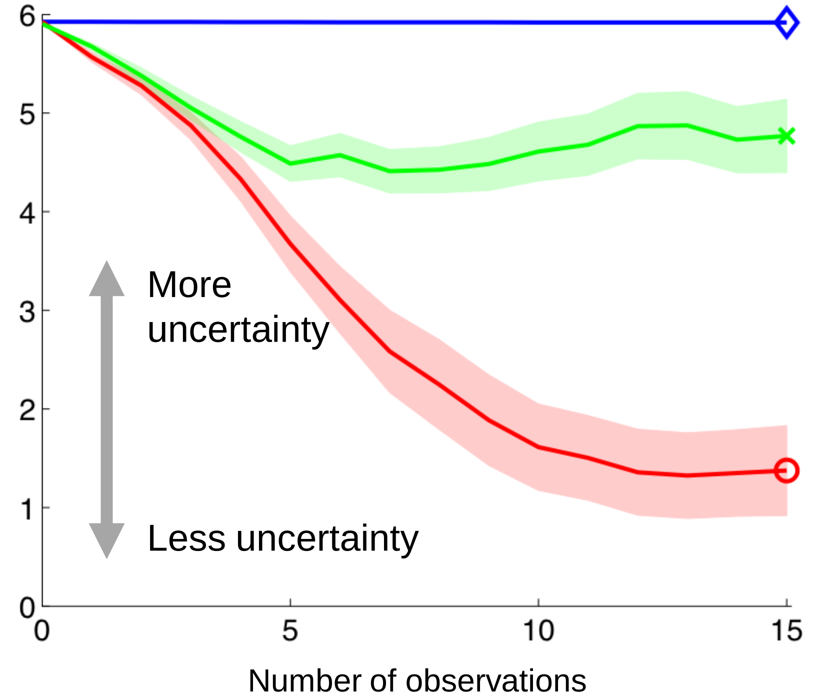
Gaussian mixture | Direct approximation | Integration wrt. q_1 | Integration wrt. q_2

Evaluation in the context of active learning

RMSD between posterior mean and x^*

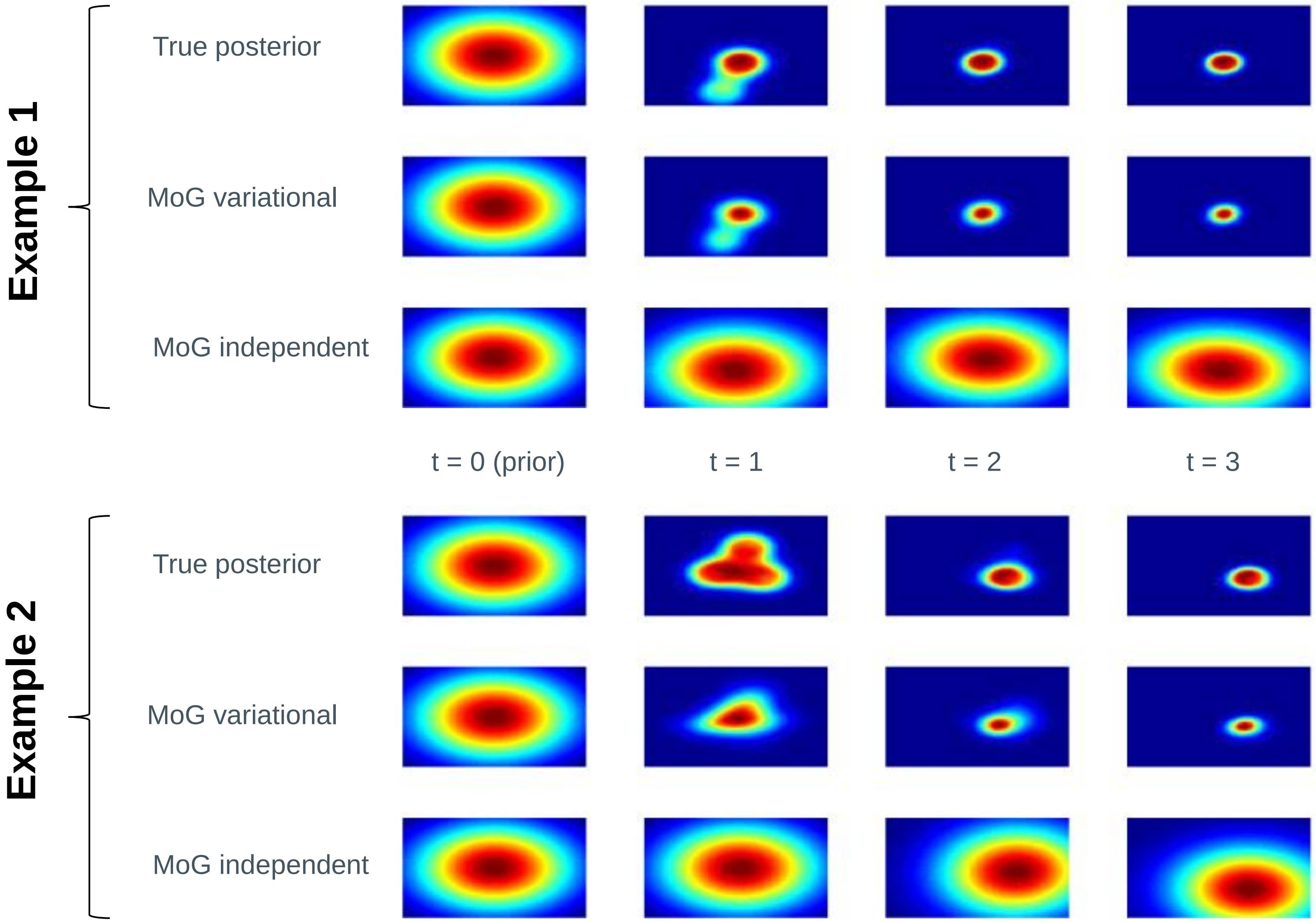


Root-determinant of posterior covariance



(state space: 10 dimensions; average over 25 runs for each approximation method)

Qualitative Results: 2D



(Observation function $f(x)$ with 3 RBF kernels in both examples)