Inverse Problems

Sommersemester 2022

Vesa Kaarnioja
vesa.kaarnioja@fu-berlin.de

FU Berlin, FB Mathematik und Informatik

Eleventh lecture, July 11, 2022

Oral exam practical matters

- The oral exam will be held on **Monday, August 1**. Please contact vesa.kaarnioja@fu-berlin.de in advance to organize a personal exam appointment time!
- The oral exam will take 30 minutes.
- The date of the make-up oral exam will be announced at a later time. The make-up oral exam will take place in early October. Please see the course page http://numerik.mi.fu-berlin.de/wiki/SS_2022/InverseProblems.php for updates.

Other practical matters

- The 11^{th} exercise sheet is the final one. There will be a "bonus" exercise sheet next week, but this will not be graded. I will be happy to give feedback on your solutions, however!
- **The final lecture will be uploaded as a recording on the course webpage on Monday 18 July. There will be no in-person lecture or exercise session on Monday 18 July.**
  - If you wish to discuss the solutions to next week's exercises, please contact `vesa.kaarnioja@fu-berlin.de` via email and/or organize an appointment after returning your solutions.

So far we have discussed inverse problems with a *static target*:

$$y = F(x) + \eta, \quad \eta \sim \mathcal{N}(0, \gamma^2 I).$$

Consider the setting where we have repeated independent observations: measure y N times, assuming that the target remains the same during the process. Let

$$\begin{aligned} D &= \{y_1, \ldots, y_N\} \quad \text{(data)}, \\ y_j &= F(x) + \eta_j. \end{aligned}$$

Likelihood:

$$\mathbb{P}(y_1, \ldots, y_N | x) \propto \prod_{j=1}^{N} \mathbb{P}(y_j | x) \propto \exp\left(-\frac{1}{2\gamma^2} \sum_{j=1}^{N} \|y_j - F(x)\|^2 \right),$$

since we assumed additive Gaussian noise.

Note that

$$\sum_{j=1}^{N} \|y_j - F(x)\|^2$$

$$= \sum_{j=1}^{N} y_j^{\mathrm{T}} y_j - 2F(x)^{\mathrm{T}} \underbrace{\sum_{j=1}^{N} y_j}_{=N\overline{y}} + N\|F(x)\|^2$$

$$= N(\|F(x)\|^2 - F(x)^{\mathrm{T}}\overline{y} + \overline{y}^{\mathrm{T}}\overline{y}) + \underbrace{N\left(\frac{1}{N}\sum_{j=1}^{N} y_j^{\mathrm{T}} y_j - \overline{y}^{\mathrm{T}}\overline{y}\right)}_{=:C}$$

$$= N\|F(x) - \overline{y}\|^2 + \underbrace{NC}_{=\text{constant}} .$$

Therefore

$$\mathbb{P}(y_1, \ldots, y_N|x) \propto \exp\left(-\frac{1}{2\gamma^2} \sum_{j=1}^{N} \|y_j - F(x)\|^2 \right)$$
$$\propto \exp\left(-\frac{1}{2(\gamma^2/N)} \|\overline{y} - F(x)\|^2 \right).$$

Hence, repeating the measurement *independently* N times is equivalent to replacing the model with

$$\overline{y} = F(x) + \eta, \qquad \eta \sim \mathcal{N}\left(0, \frac{\gamma^2}{N}\right).$$

Variance reduction of the noise.

It is essential that the target does not change during the measurement process.

Examples where the condition may not be valid:

- EEG
- Target tracking
- Weather forecasting

Dynamic inverse problems

More general observation model:

$$y_j = F(x_j) + \eta_j, \quad j = 1, 2, \ldots, J.$$

The observations cannot be integrated unless we have a *dynamic prior model*.

One of the simplest dynamic prior models is a 1-Markov evolution model

$$x_{j+1} = G(x_j) + \xi_{j+1}, \quad j = 0, 1, \ldots, J-1,$$

where $G : \mathbb{R}^d \to \mathbb{R}^d$ is presumably known and ξ_{j+1} is an *innovation process*.

Examples

- Static measurement: $G(x) = x$, $\xi_{j+1} = 0$.
- Random walk model (often used in lack of anything more sophisticated):

$$x_{j+1} = x_j + \xi_{j+1}, \quad \xi_{j+1} \sim \mathcal{N}(0, \sigma^2 I).$$

- First order differential equation: assume that the unknown is a time-dependent vector $x(t) \in \mathbb{R}^d$ satisfying *ideally* the differential equation

$$x'(t) = f(x(t), t).$$

Time discretization: let $t_j = jh$, $j = 0, 1, \ldots$, and write $x_j = x(t_j)$. Then we can use finite differences, e.g., forward Euler method

$$x_{j+1} = x_j + hf(x_j, t_j) + \xi_{j+1}$$

or backward Euler method

$$x_{j+1} = x_j + hf(x_{j+1}, t_{j+1}) + \xi_{j+1},$$

where ξ_{j+1} accounts for discretization errors as well as possible deviations from the ideal.

Basic form of Bayes filtering

Evolution-observation model:

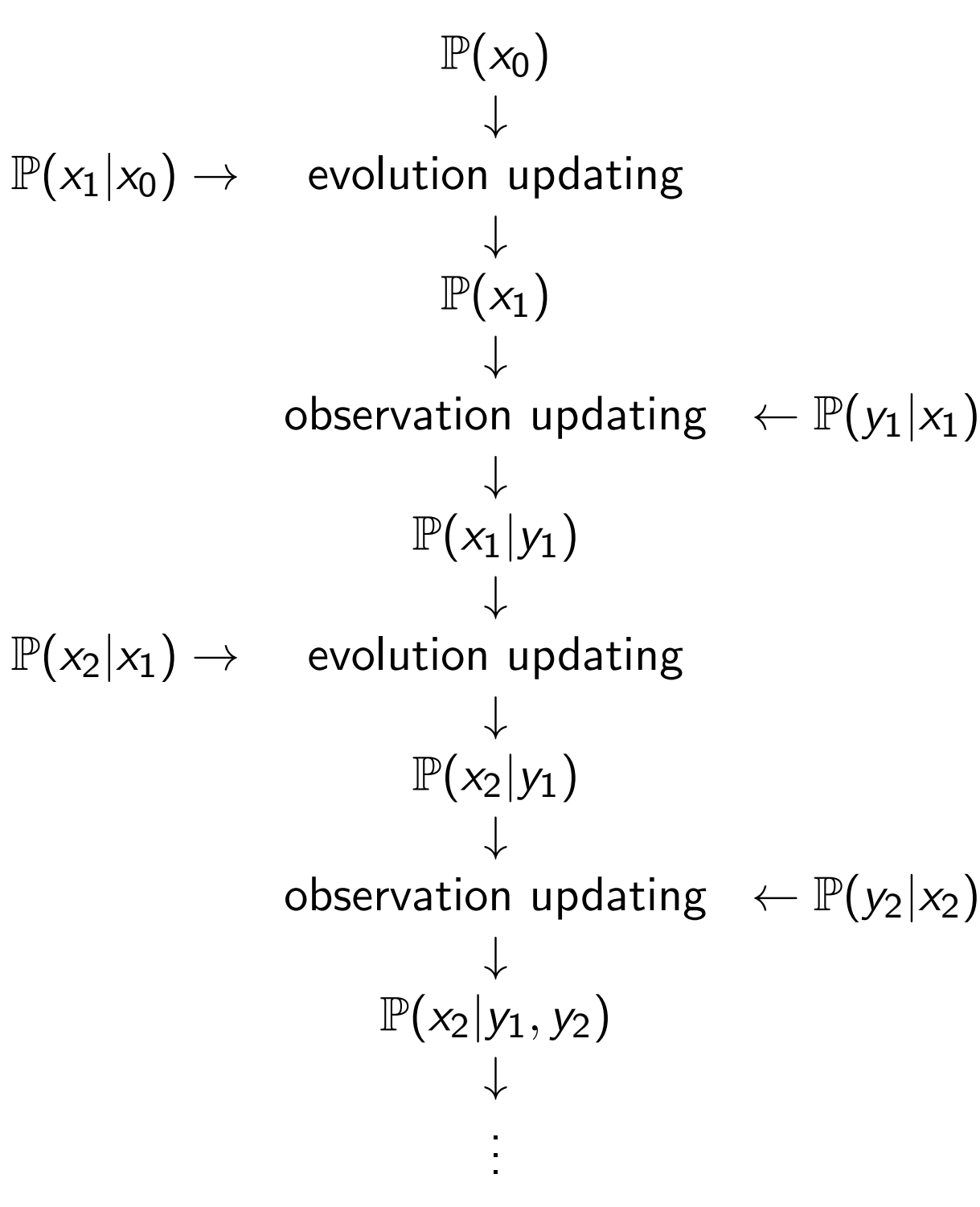
$$x_{j+1} = G(x_j) + \xi_{j+1}, \quad j = 0, 1, \ldots, J-1,$$
$$y_{j+1} = F(x_{j+1}) + \eta_{j+1}, \quad j = 0, 1, \ldots, J-1.$$

The observations $y_1, \ldots, y_J$ and the prior probability density of x_0 are given.

Adaptive algorithm

The goal is an algorithm which works as follows:

- Given the density of x_0, *predict* the density of x_1 using the prior evolution model.
- Using the predicted density of x_1 as *prior*, calculate the posterior density of $x_1|y_1$.
- Using the posterior density of $x_1|y_1$, predict the density of x_2.
- Using the predicted density of x_2 as *prior*, calculate the posterior density of $x_2|y_1, y_2$.
- Continue similarly.


$\mathbb{P}(x_0)$
$\mathbb{P}(x_1|x_0) \rightarrow$ evolution updating
$\mathbb{P}(x_1)$
observation updating $\leftarrow \mathbb{P}(y_1|x_1)$
$\mathbb{P}(x_1|y_1)$
$\mathbb{P}(x_2|x_1) \rightarrow$ evolution updating
$\mathbb{P}(x_2|y_1)$
observation updating $\leftarrow \mathbb{P}(y_2|x_2)$
$\mathbb{P}(x_2|y_1, y_2)$
$\vdots$

- **Prediction step:** Given the density of x_j, calculate the density of x_{j+1} from

$$x_{j+1} = G(x_j) + \xi_{j+1}. \qquad \text{(propagation problem)}$$

- **Correction step:** Given the prior density of x_{j+1}, calculate the posterior density of $x_{j+1}|y_{j+1}$ using the observational model

$$y_{j+1} = F(x_{j+1}) + \eta_{j+1}. \qquad \text{(inverse problem)}$$

Particular approaches

- Linear model, Gaussian innovation and error: classical Kalman filtering.
- Linearization of non-linear evolution (or observation) model: extended Kalman filtering.
- Nonlinear and/or non-Gaussian models: particle filtering.

Kalman filter

Consider the linear ($G(\cdot) = M\cdot$, $F(\cdot) = H\cdot$) evolution-observation system

$$x_{j+1} = Mx_j + \xi_{j+1}, \quad \xi_{j+1} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Sigma),$$
$$y_{j+1} = Hx_{j+1} + \eta_{j+1}, \quad \eta_{j+1} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Gamma).$$

Prediction: Suppose $x_j \sim \mathcal{N}(m_j, C_j)$. Then (cf. exercise from week 5)

$$x_{j+1} = Mx_j + \xi_{j+1} \sim \mathcal{N}(\widehat{m}_{j+1}, \widehat{C}_{j+1}),$$

where $\widehat{m}_{j+1} = Mm_j$ and $\widehat{C}_{j+1} = MC_jM^{\mathrm{T}} + \Sigma$.

Correction: Linear Gaussian setting implies $x_{j+1}|y_{j+1} \sim \mathcal{N}(m_{j+1}, C_{j+1})$, where (cf. exercise from week 7)

$$m_{j+1} = \widehat{m}_{j+1} + \widehat{C}_{j+1}H^{\mathrm{T}}(H\widehat{C}_{j+1}H^{\mathrm{T}} + \Gamma)^{-1}(y_{j+1} - H\widehat{m}_{j+1}),$$
$$C_{j+1} = \widehat{C}_{j+1} - \widehat{C}_{j+1}H^{\mathrm{T}}(H\widehat{C}_{j+1}H^{\mathrm{T}} + \Gamma)^{-1}H\widehat{C}_{j+1}.$$

Remark: The expensive step in Kalman filtering is the computation of the so-called *Kalman gain* matrix:

$$K_{j+1} = \widehat{C}_{j+1}H^{\mathrm{T}}(H\widehat{C}_{j+1}H^{\mathrm{T}} + \Gamma)^{-1}.$$

Kalman filter algorithm

Given: Initial distribution for $x_0 \sim \mathcal{N}(m_0, C_0)$, where $m_0 \in \mathbb{R}^d$ and $C_0 \in \mathbb{R}^{d \times d}$ is symmetric and positive definite.

for $j = 0, 1, 2, \ldots, J-1$, **do**

Prediction step:

$$\widehat{m}_{j+1} = M m_j$$
$$\widehat{C}_{j+1} = M C_j M^{\mathrm{T}} + \Sigma$$

Correction step:

$$K_{j+1} = \widehat{C}_{j+1} H^{\mathrm{T}} (H \widehat{C}_{j+1} H^{\mathrm{T}} + \Gamma)^{-1}$$
$$m_{j+1} = \widehat{m}_{j+1} + K_{j+1}(y_{j+1} - H \widehat{m}_{j+1})$$
$$C_{j+1} = \widehat{C}_{j+1} - K_{j+1} H \widehat{C}_{j+1}$$

end for

Output: Predicted distributions for $x_j \sim \mathcal{N}(\widehat{m}_{j+1}, \widehat{C}_{j+1})$ and filtering distributions for $x_{j+1} | y_1, \ldots, y_{j+1} \sim \mathcal{N}(m_{j+1}, C_{j+1})$, $j = 0, \ldots, J-1$.

Extended Kalman filter (non-linear evolution model)

Consider non-linear $G\colon \mathbb{R}^d \to \mathbb{R}^d$ and linear $F(\cdot) = H\cdot$ with

$$x_{j+1} = G(x_j) + \xi_{j+1}, \quad \xi_{j+1} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Sigma),$$
$$y_{j+1} = Hx_{j+1} + \eta_{j+1}, \quad \eta_{j+1} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Gamma),$$

with $x_0 \sim \mathcal{N}(m_0, C_0)$.

Prediction: Suppose $x_j \sim \mathcal{N}(m_j, C_j)$. We can linearize

$$x_{j+1} = G(x_j) + \xi_{j+1} \approx G(m_j) + DG(m_j)(x_j - m_j) + \xi_j.$$

An affine transformation is still Gaussian, so we obtain the approximations

$$\widehat{m}_{j+1} = G(m_j), \ \widehat{C}_{j+1} = DG(m_j)C_jDG(m_j)^{\mathrm{T}} + \Sigma.$$

Correction: Now that $x_{j+1} \sim \mathcal{N}(\widehat{m}_{j+1}, \widehat{C}_{j+1})$, we can use the linear Gaussian setting to obtain $x_{j+1}|y_{j+1} \sim \mathcal{N}(m_{j+1}, C_{j+1})$ with

$$m_{j+1} = \widehat{m}_{j+1} + \widehat{C}_{j+1}H^{\mathrm{T}}(H\widehat{C}_{j+1}H^{\mathrm{T}} + \Gamma)^{-1}(y_{j+1} - B\widehat{m}_{j+1}),$$
$$C_{j+1} = \widehat{C}_{j+1} - \widehat{C}_{j+1}H^{\mathrm{T}}(H\widehat{C}_{j+1}H^{\mathrm{T}} + \Gamma)^{-1}H\widehat{C}_{j+1}.$$

Ensemble Kalman filter (non-linear evolution model)

Consider

$$x_{j+1} = G(x_j) + \xi_{j+1}, \quad \xi_{j+1} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Sigma),$$
$$y_{j+1} = Hx_{j+1} + \eta_{j+1}, \quad \eta_{j+1} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Gamma),$$

with $x_0 \sim \mathcal{N}(m_0, C_0)$.

The computation of the analytical predictive covariances and (in the non-linear setting) the Jacobi matrix become computationally inefficient and expensive for high-dimensional systems. The basic idea of ensemble Kalman filter is as follows:

1. Draw a sample from the initial distribution of x_0 ("initial ensemble")
2. Replace the predictive mean $\widehat{m}_{j+1}$ and covariance $\widehat{C}_{j+1}$ as well as the filtering mean m_{j+1} and covariance C_{j+1} with their corresponding sample means and covariances by propagating the initial ensemble through the evolution-observation model.

Ensemble Kalman filter algorithm

Given: Ensemble size N. Initial ensemble $\{x_0^{(i)}\}_{i=1}^N$ drawn from the initial distribution of $x_0 \sim \mathcal{N}(m_0, C_0)$, where $m_0 \in \mathbb{R}^d$ and $C_0 \in \mathbb{R}^{d\times d}$ is symmetric and positive definite. Parameter $s \in \{0, 1\}$.

for $j = 0, 1, 2, \ldots, J-1$**, do**

Prediction step:

$$\text{draw } \xi_{j+1}^{(i)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Sigma), \ i = 1, \ldots, N,$$

$$\widehat{x}_{j+1}^{(i)} = G(x_j^{(i)}) + \xi_{j+1}^{(i)}, \quad i = 1, \ldots, N,$$

$$\widehat{m}_{j+1} = \frac{1}{N}\sum_{i=1}^{N} \widehat{x}_{j+1}^{(i)} \quad \text{and} \quad \widehat{C}_{j+1} = \frac{1}{N}\sum_{i=1}^{n} (\widehat{x}_{j+1}^{(i)} - \widehat{m}_{j+1})(\widehat{x}_{j+1}^{(i)} - \widehat{m}_{j+1})^{\mathrm{T}}.$$

Correction step:

$$\text{draw } \eta_{j+1}^{(i)} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Gamma), \quad i = 1, \ldots, N,$$

$$y_{j+1}^{(i)} = y_{j+1} + s\eta_{j+1}^{(i)}, \quad i = 1, \ldots, N,$$

$$K_{j+1} = \widehat{C}_{j+1} H^{\mathrm{T}} (H \widehat{C}_{j+1} H^{\mathrm{T}} + \Gamma)^{-1},$$

$$x_{j+1}^{(i)} = \widehat{x}_{j+1}^{(i)} + K_{j+1}(y_{j+1}^{(i)} - H\widehat{x}_{j+1}^{(i)}), \quad i = 1, \ldots, N.$$

end for

Output: Ensembles $\{x_j^{(i)}\}_{i=1}^N$, $j = 0, \ldots, J$.

Remark:

- Setting the parameter $s = 1$ is suitable at approximating the Kalman filter in linear Gaussian settings: if each prediction particle $\widetilde{x}_{j+1}^{(i)}$ is distributed according to a non-degenerate Gaussian distribution, then in the linear Gaussian setting the "corrected" particle $x_{j+1}^{(i)}$ will be Gaussian with mean and covariance that agree with the usual Kalman filter formulae. See Theorem 10.1 in Sanz-Alonso, Stuart, Taeb 2018.
- Setting the parameter $s = 0$ is natural if viewing the algorithm as a sequential optimizer in problems where the filtering distributions are not well approximated by Gaussians. See Section 8.1.2 in Sanz-Alonso, Stuart, Taeb 2018.

General evolution-observation model and particle filters

Consider the more general model

$$x_{j+1} = G(x_j, \xi_{j+1}), \quad j = 0, 1, \ldots, J-1,$$
$$y_{j+1} = F(x_{j+1}, \eta_{j+1}), \quad j = 0, 1, \ldots, J-1.$$

The functions F and G are assumed to be known. We also assume that $\xi_{j+1} \perp x_j$ and $\eta_{j+1} \perp x_{j+1}$.

Observation and evolution models may be cumbersome or impossible to linearize (e.g., non-differentiable or no closed form). One may try Monte Carlo methods to simulate the distributions by random samples.

The goal in *particle filter* methods is to produce sequentially an ensemble of random samples $\{x_j^{(1)}, \ldots, x_j^{(N)}\}$ distributed according to the conditional probability distributions $\mathbb{P}(x_{j+1}|y_1, \ldots, y_j)$ (prediction) or $\mathbb{P}(x_j|y_1, \ldots, y_j)$ (filtering). The vectors $x_j^{(i)}$ are called *particles* of the sample, hence the name particle filter.

One straightforward particle filter method is known as the *sampling importance resampling* filter (also known as *SIR* or *bootstrap filter*).

Sampling importance resampling (Bootstrapping)

1. Set $j = 0$ and generate an initial sample $S_0 = \{x_0^{(i)}\}_{i=1}^N$ by drawing from the density $\mathbb{P}(x_0)$. (This may require MCMC if the initial density is complicated, e.g., non-Gaussian.)
2. **Prediction:** Draw $\xi_{j+1}^{(i)}$ from the distribution of ξ_{j+1} and set $\widehat{x}_{j+1}^{(i)} = G(x_j^{(i)}, \xi_{j+1}^{(i)})$ for $1 \le i \le N$. Let $\widehat{S}_{j+1} = \{\widehat{x}_{j+1}^{(i)}\}_{i=1}^N$.
3. **Correction:** Assume that from the observational model $y_j = F(x_j, \eta_j)$, we can calculate the likelihood density $C\mathbb{P}(y_j|x_j)$, $j = 1, 2, \ldots, J$, up to a multiplicative constant $C > 0$.[†] Calculate the importance of each propagated particle

$$\widehat{w}_{j+1}^{(i)} = C\mathbb{P}(y_{j+1}|\widehat{x}_{j+1}^{(i)}), \quad 1 \le i \le N,$$

and compute their *relative importance*

$$w_{j+1}^{(i)} = \frac{\widehat{w}_{j+1}^{(i)}}{W}, \quad W = \sum_{i=1}^{N} \widehat{w}_{j+1}^{(i)}.$$

Resampling: draw a new sample $S_{j+1} = \{x_{j+1}^{(1)}, \ldots, x_{j+1}^{(N)}\}$ from the sample $\widehat{S}_{j+1}$, with the probability of drawing $\widehat{x}_{j+1}^{(i)}$ set equal to $w_{j+1}^{(i)}$. Set $j \leftarrow j + 1$ and return to step 2.

[†] E.g., if $y_j = F(x_j) + \eta_j$, $\eta_j \sim \mathcal{N}(0, \Gamma)$, then $\mathbb{P}(y_j|x_j) \propto \exp(-\frac{1}{2}\|y_j - F(x_j)\|_{\Gamma^{-1}}^2)$.

Numerical example

Let us consider the heat equation with insulating boundary conditions

$$\begin{cases} \partial_t u(x,t) = \partial_x^2(x,t), & x \in (0,1),\ t \in (0,T), \\ \partial_x u(0,t) = \partial_x(1,t) = 0, & t \in (0,T), \\ u(x,0) = f(x), & x \in (0,1), \end{cases}$$

where $f : [0,1] \to \mathbb{R}$ is a (poorly known) initial heat distribution.

The temperature is measured at discrete times $t_j = j\Delta t$ at the end points:

$$y_j = \begin{bmatrix} u(0,t_j) \\ u(1,t_j) \end{bmatrix} + \eta_j, \quad j \in \{1,\dots,J\},$$

where $\eta_j \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0,\gamma^2 I_2)$.

Goal: Track the target heat distribution $u(x,t)$ for $t \in \{t_1,\dots,t_J\}$ and $x \in (0,1)$.

Discretization of the model problem

Divide the spatial interval into equally spaced subintervals with end points $x_j = j/n$, $0 \le j \le n$. Further, let $\boldsymbol{u}(t) := [u(x_1, t), \ldots, u(x_{n-1}, t)]^{\mathrm{T}}$. We have

$$\partial_x^2 u(x_j, t) \approx n^2 (u(x_{j-1}, t) - 2u(x_j, t) + u(x_{j+1}, t)), \quad 1 \le j \le n-1.$$

The Neumann boundary conditions are approximated by

$$u(0, t) = u(x_1, t) \quad \text{and} \quad u(1, t) = u(x_{n-1}, t).$$

This yields the semidiscretized evolution equation

$$\frac{\partial}{\partial t}\boldsymbol{u}(t) = n^2 \underbrace{\begin{bmatrix} -1 & 1 & & & \\ 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & & -2 & 1 \\ & & & 1 & -1 \end{bmatrix}}_{=:L \in \mathbb{R}^{(n-1)\times(n-1)}} \boldsymbol{u}(t).$$

We simulate the measurements y_j by solving the PDE for each time point $t_j = j\Delta t$, $j = 1, \ldots, J$. To avoid the inverse crime, the forward problem is solved using a dense computational grid with spatial discretization computed using $n = 150$ grid points. The values at the boundary $x \in \{0, 1\}$ are contaminated with simulated mean-zero Gaussian measurement noise with standard deviation $\gamma = 0.0001$. The actual inversion takes place on a computational grid with $n = J = 100$. As the analytical initial heat distribution, we use $f(x) = x^2(1-x)^2$.
For Kalman filtering, we suppose that it is known that the temperature distribution follows the equation

$$\frac{\partial}{\partial t}\boldsymbol{u}(t) = L\boldsymbol{u}(t).$$

We discretize the evolution model using the backward Euler method:

$$\boldsymbol{u}(t_{j+1}) = (I - \Delta t L)^{-1}\boldsymbol{u}(t_j) + \xi_j,$$

where the innovation term is modeled as $\xi_j \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2 I)$ with $\sigma = 0.01$. Note that, in terms of the notation introduced previously, the evolution matrix is $M = (I - \Delta t L)^{-1}$ and the observation operator is $H = \begin{bmatrix} \boldsymbol{e}_1^{\mathrm{T}} \\ \boldsymbol{e}_{n-1}^{\mathrm{T}} \end{bmatrix}$.

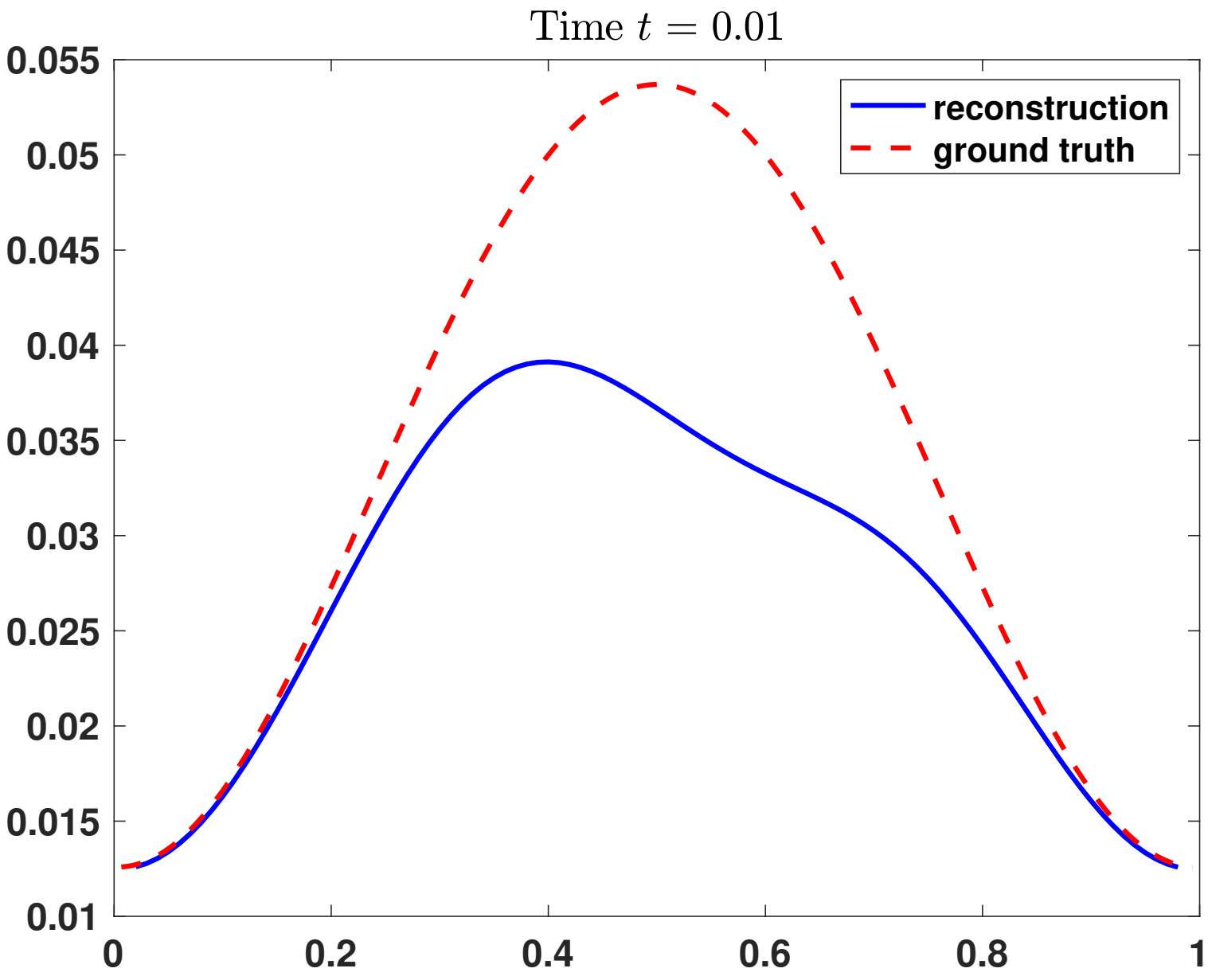
Finally, we assume that the initial heat distribution is poorly known and modeled using a Gaussian smoothness prior

$$\mathcal{N}(0, C_0),$$

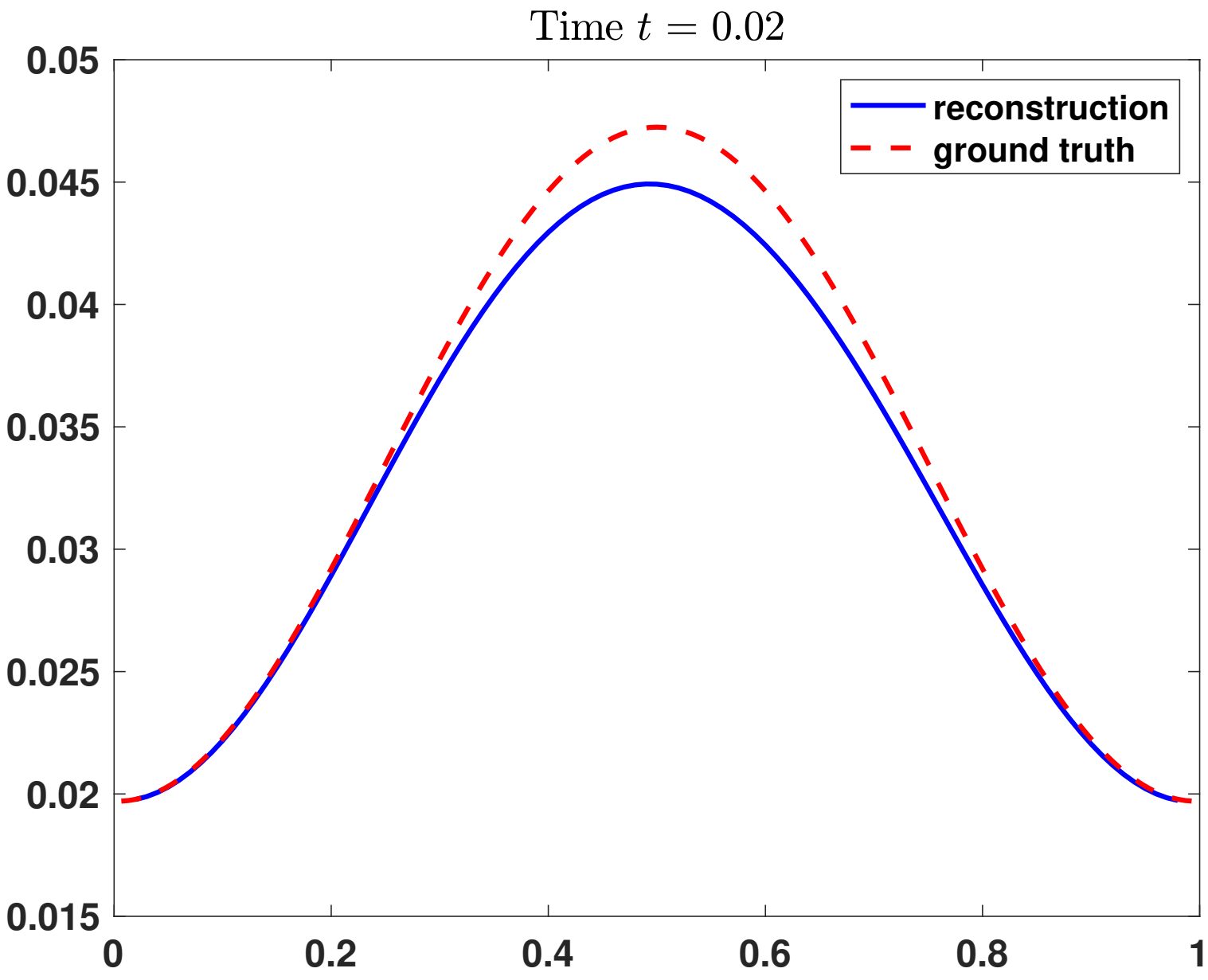
where $C_0 = 0.1(D^{\mathrm{T}}D)^{-1}$ with $D = n^{-2}L$.

We track the temperature distribution using both Kalman filtering and ensemble Kalman filtering. We display the filtered CM estimates for both approaches.

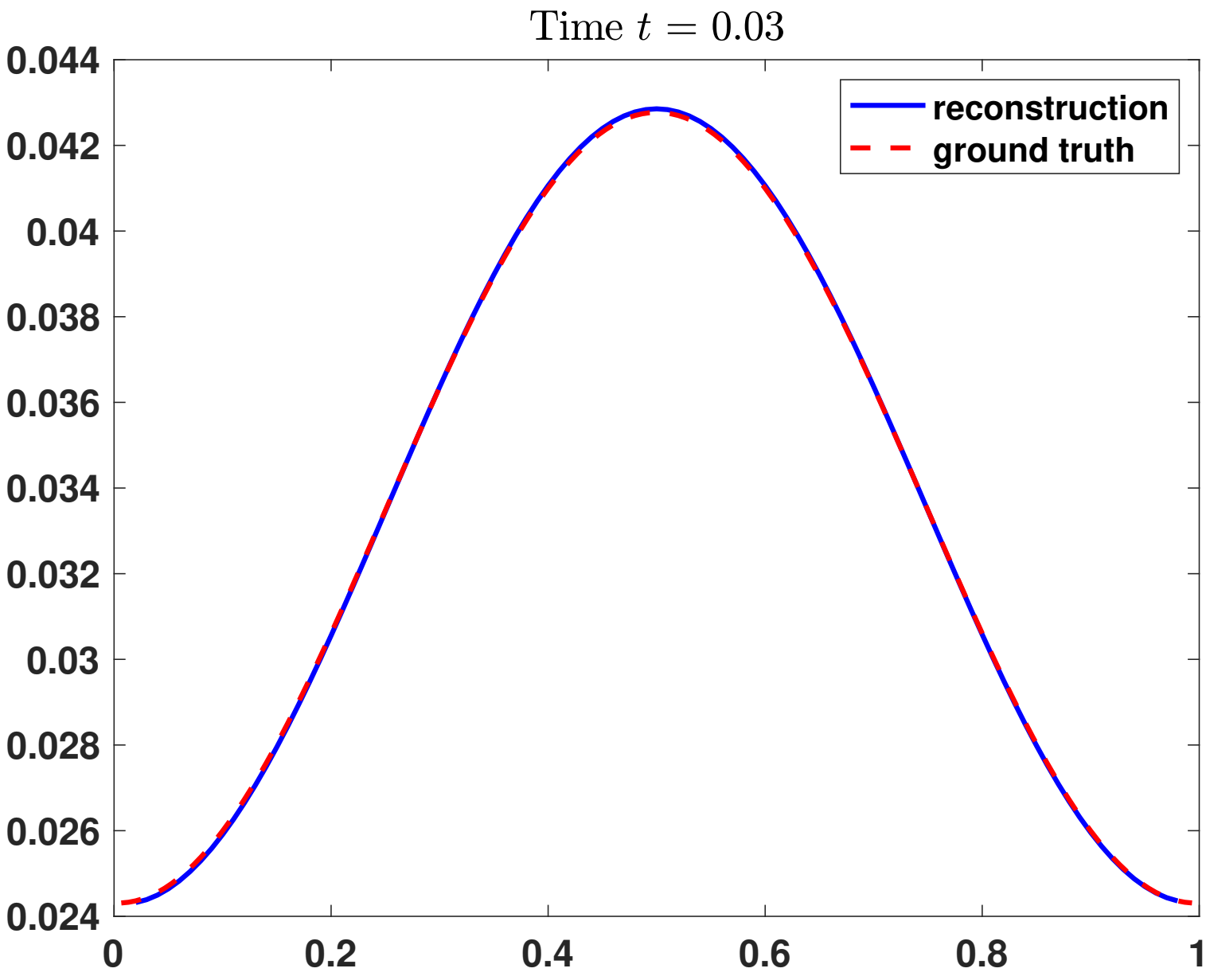
Kalman filter



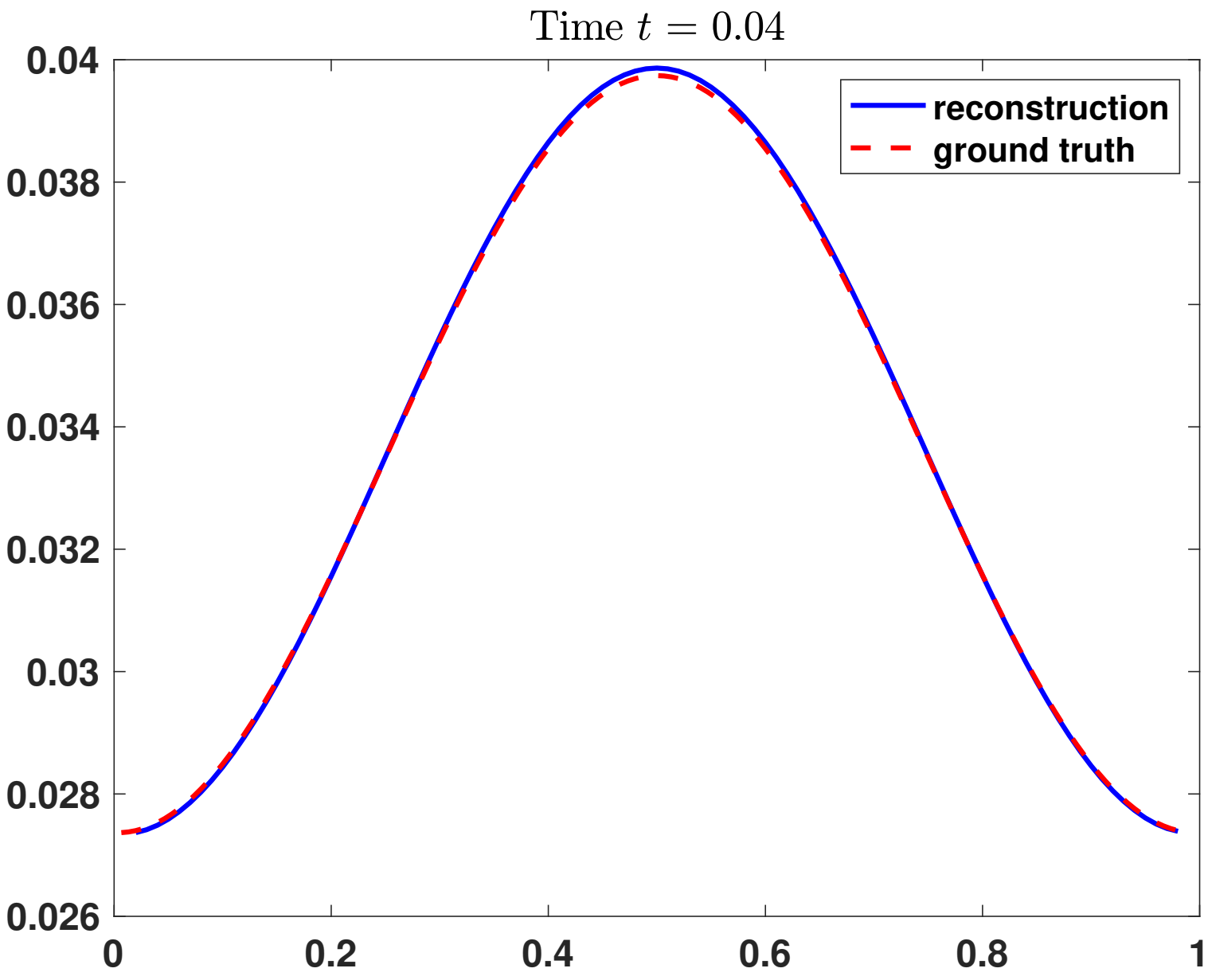
Kalman filter



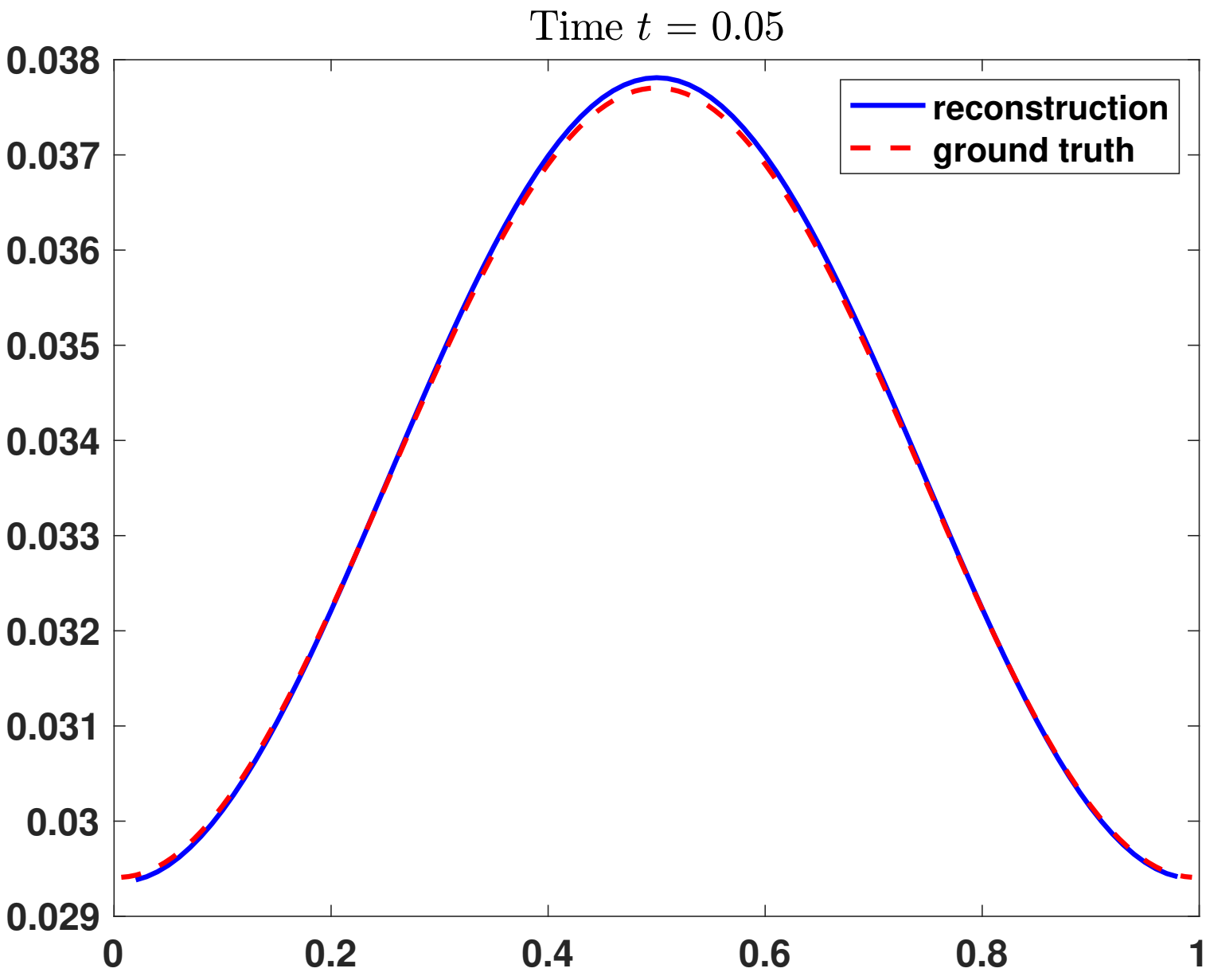
Kalman filter



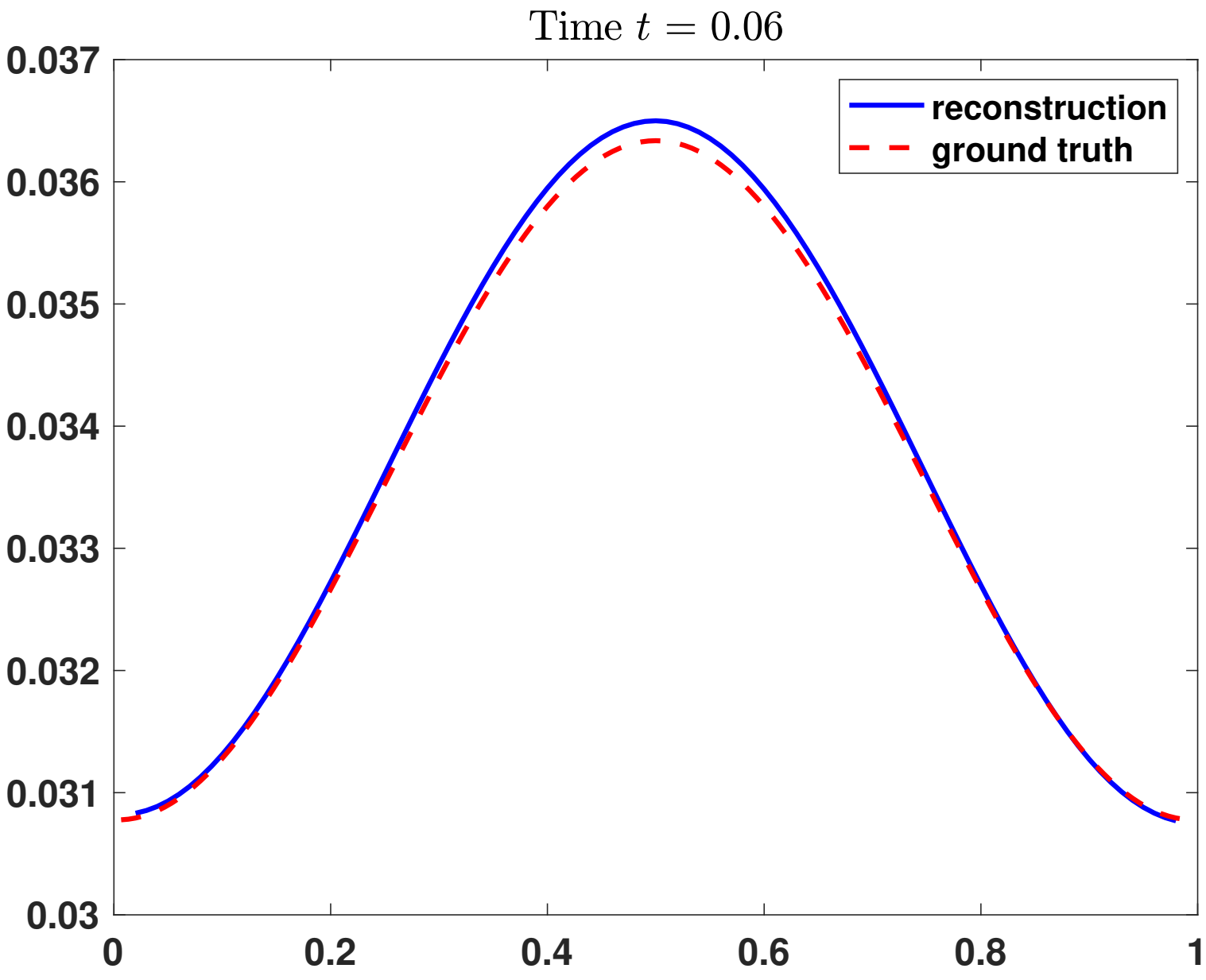
Kalman filter



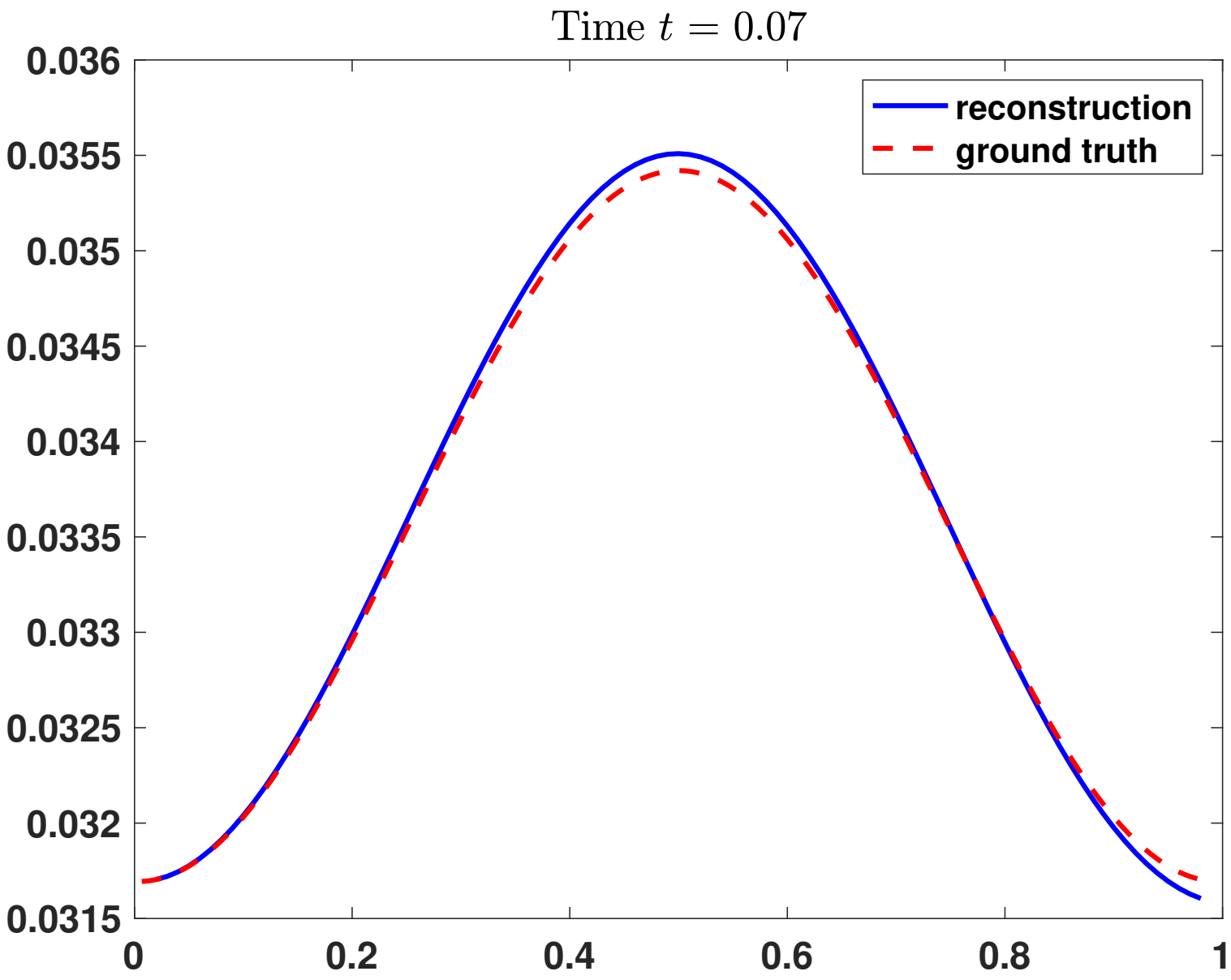
Kalman filter



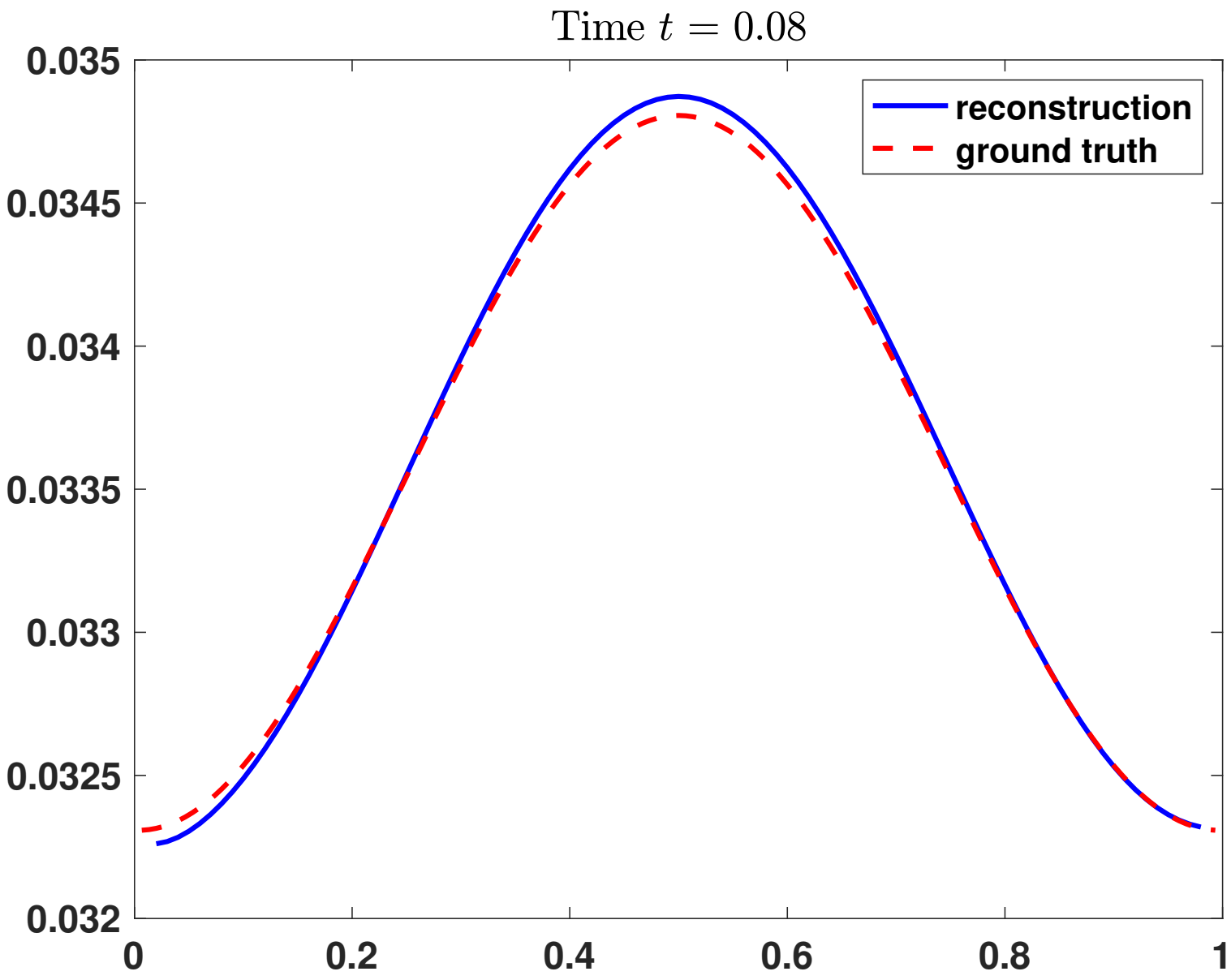
Kalman filter



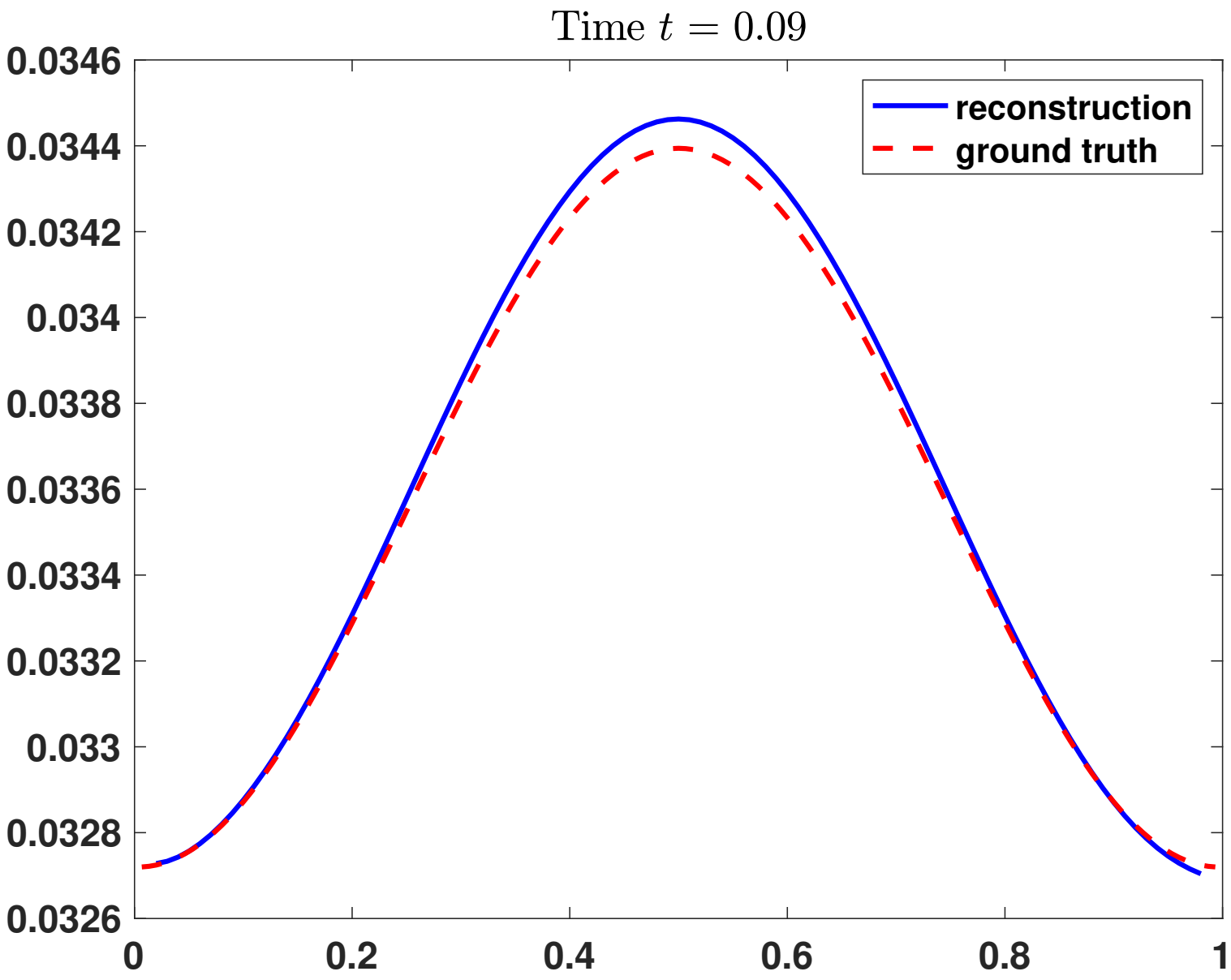
Kalman filter



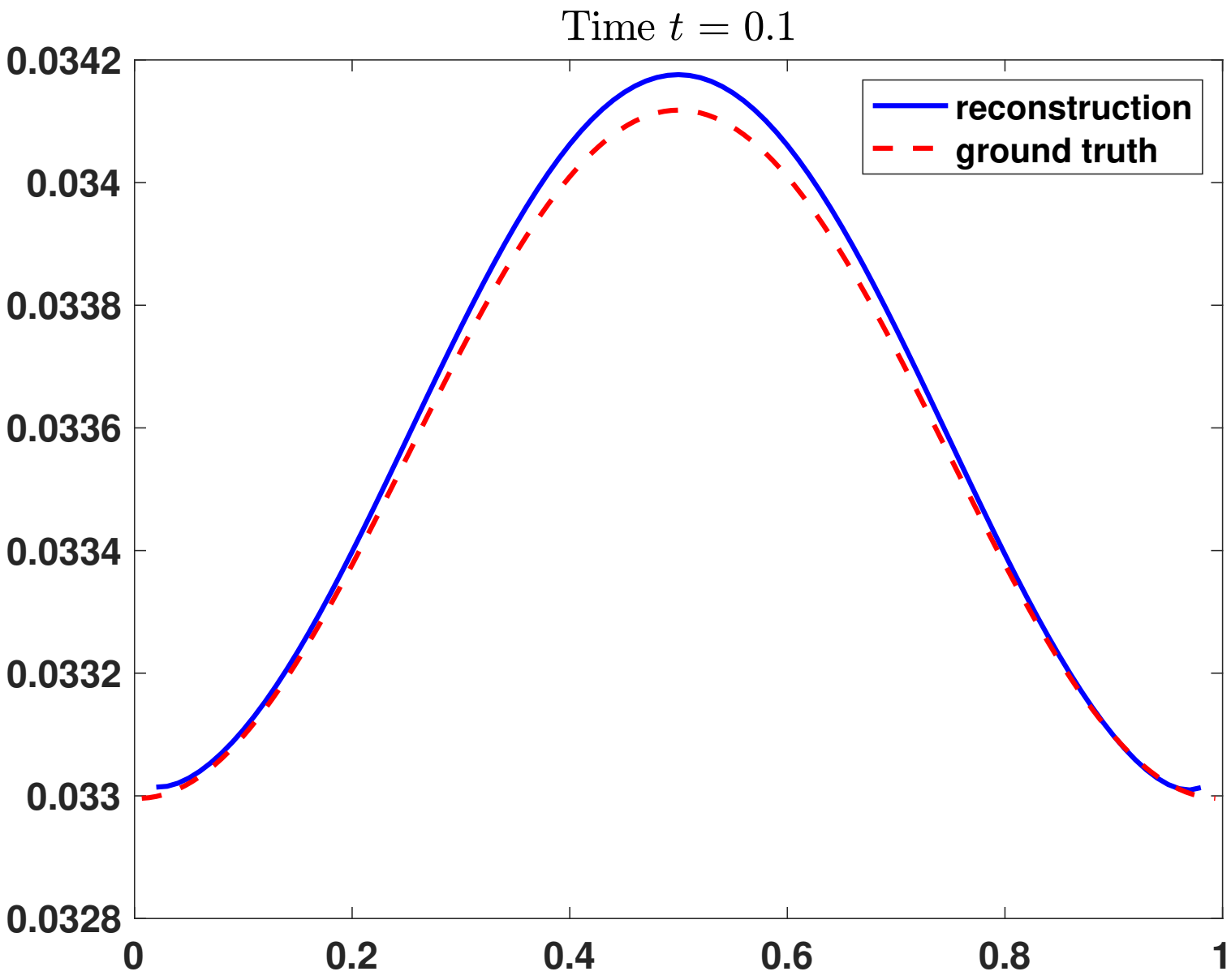
Kalman filter



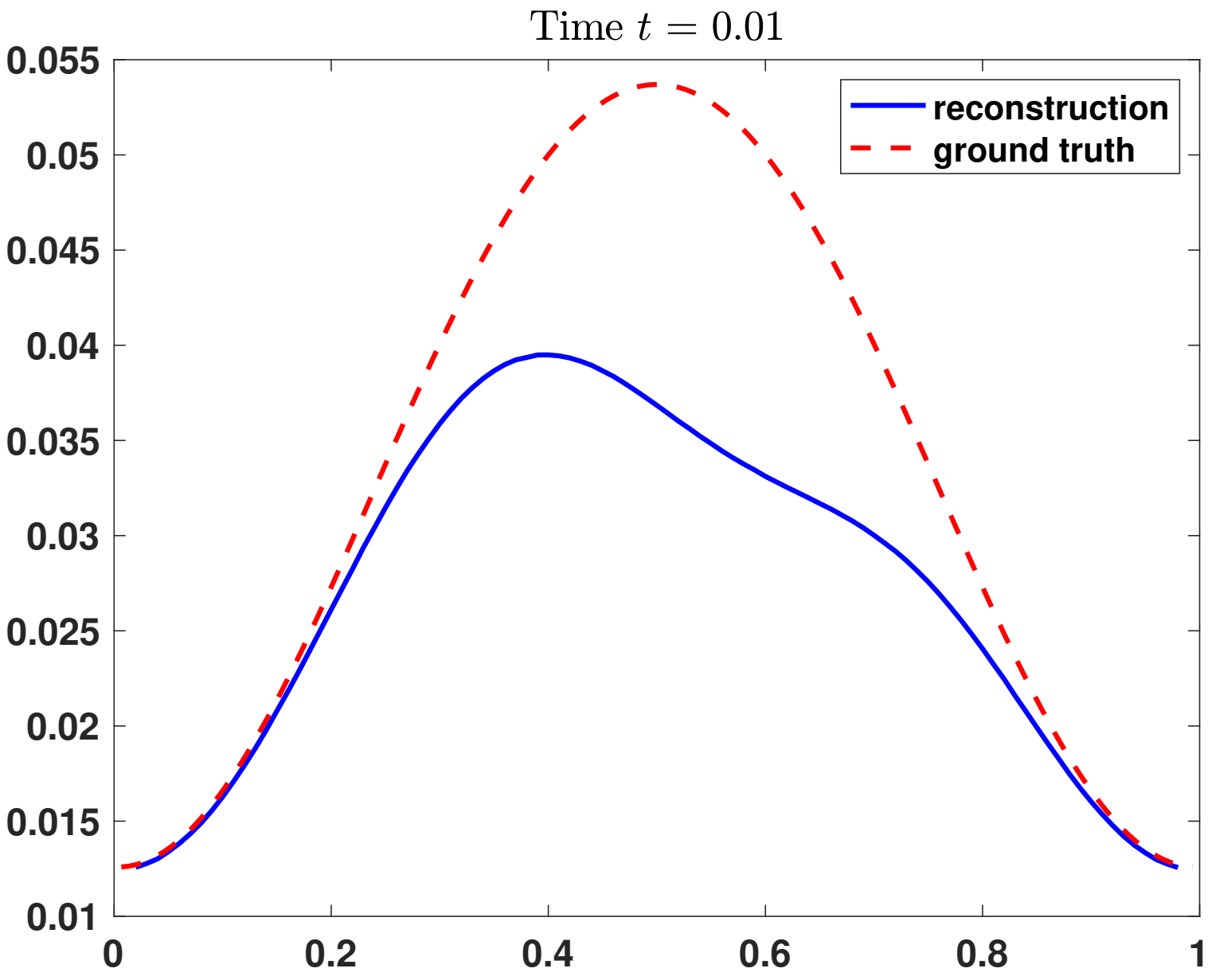
Kalman filter



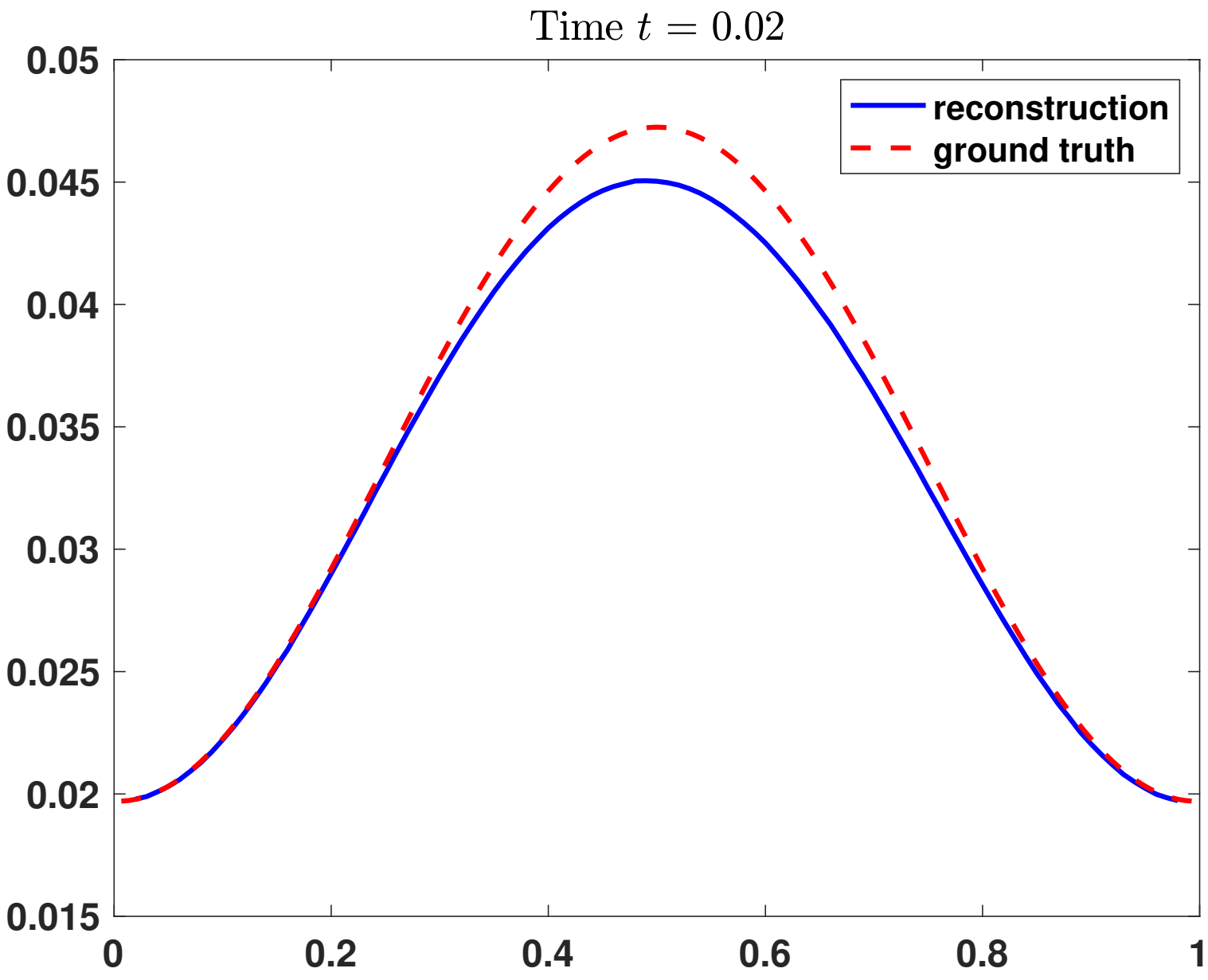
Kalman filter



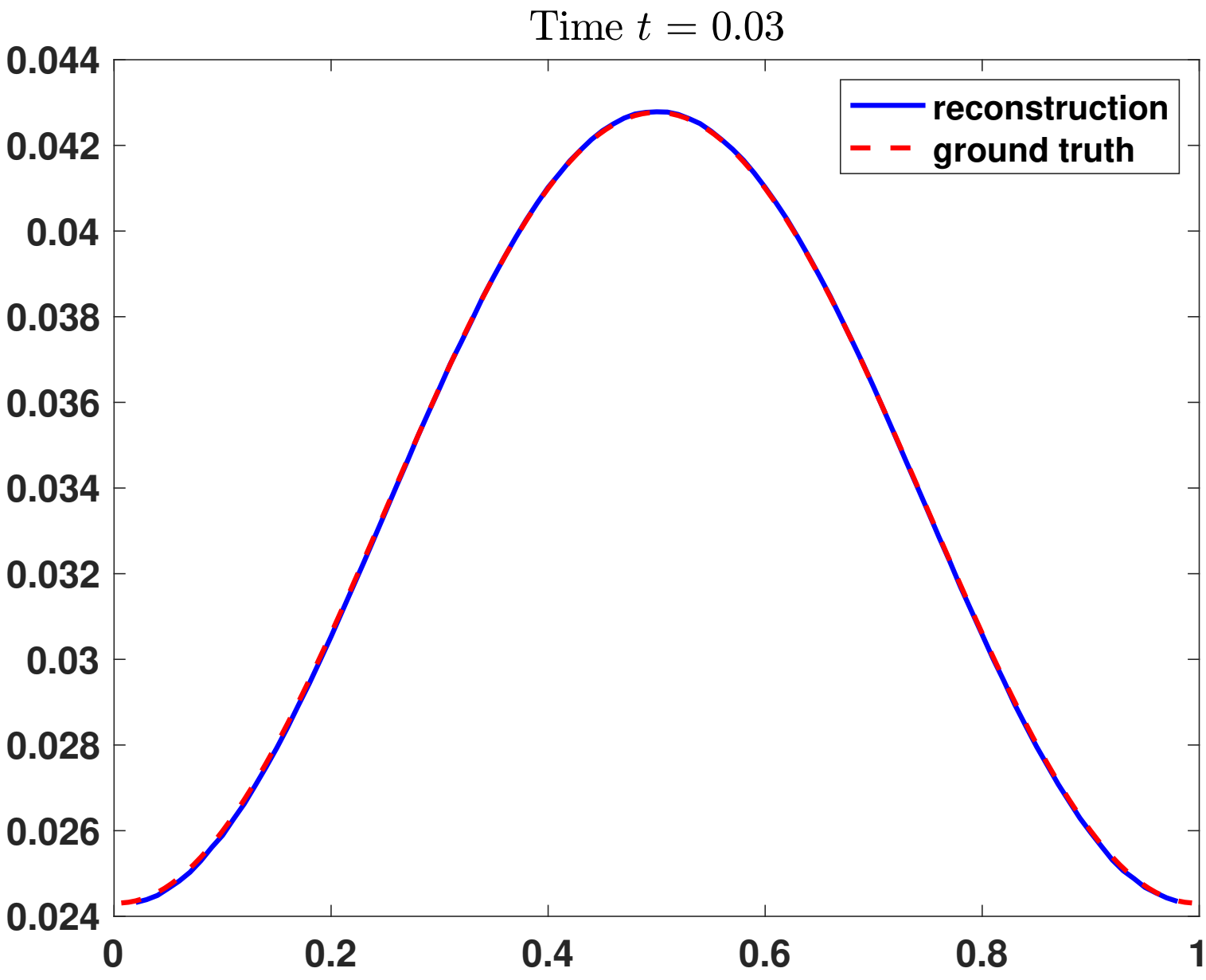
Ensemble Kalman filter ($N = 10^6$)



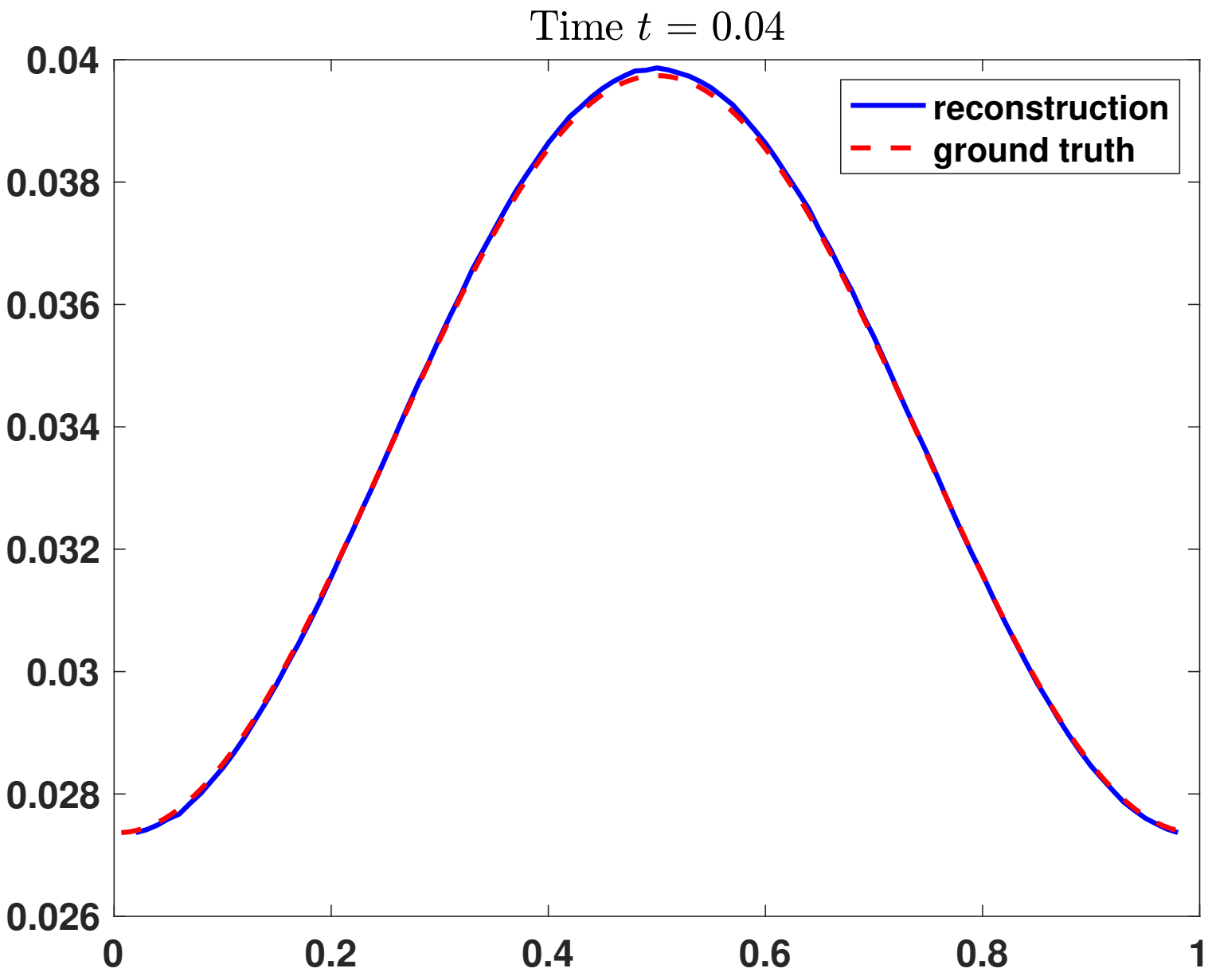
Ensemble Kalman filter ($N = 10^6$)



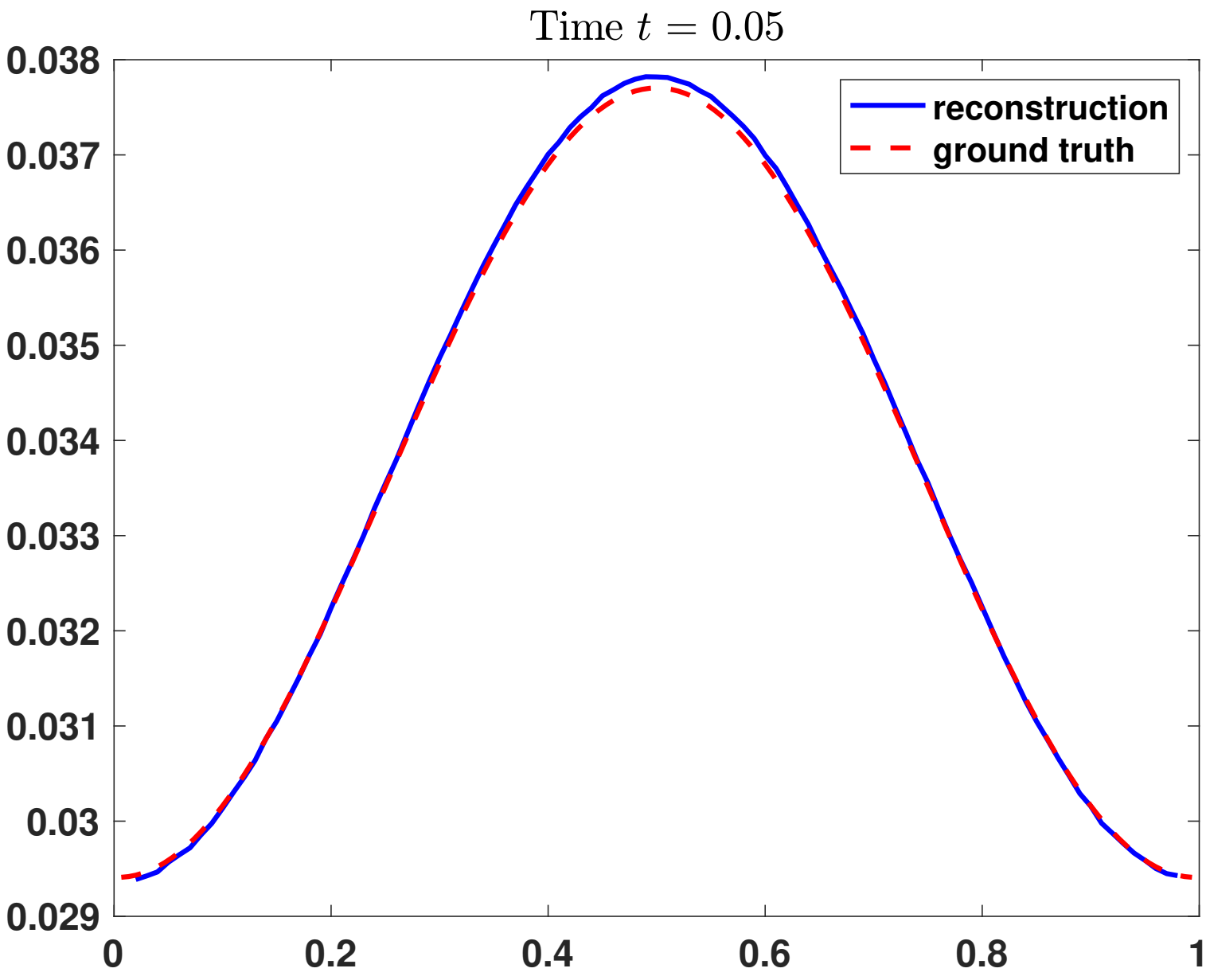
Ensemble Kalman filter ($N = 10^6$)



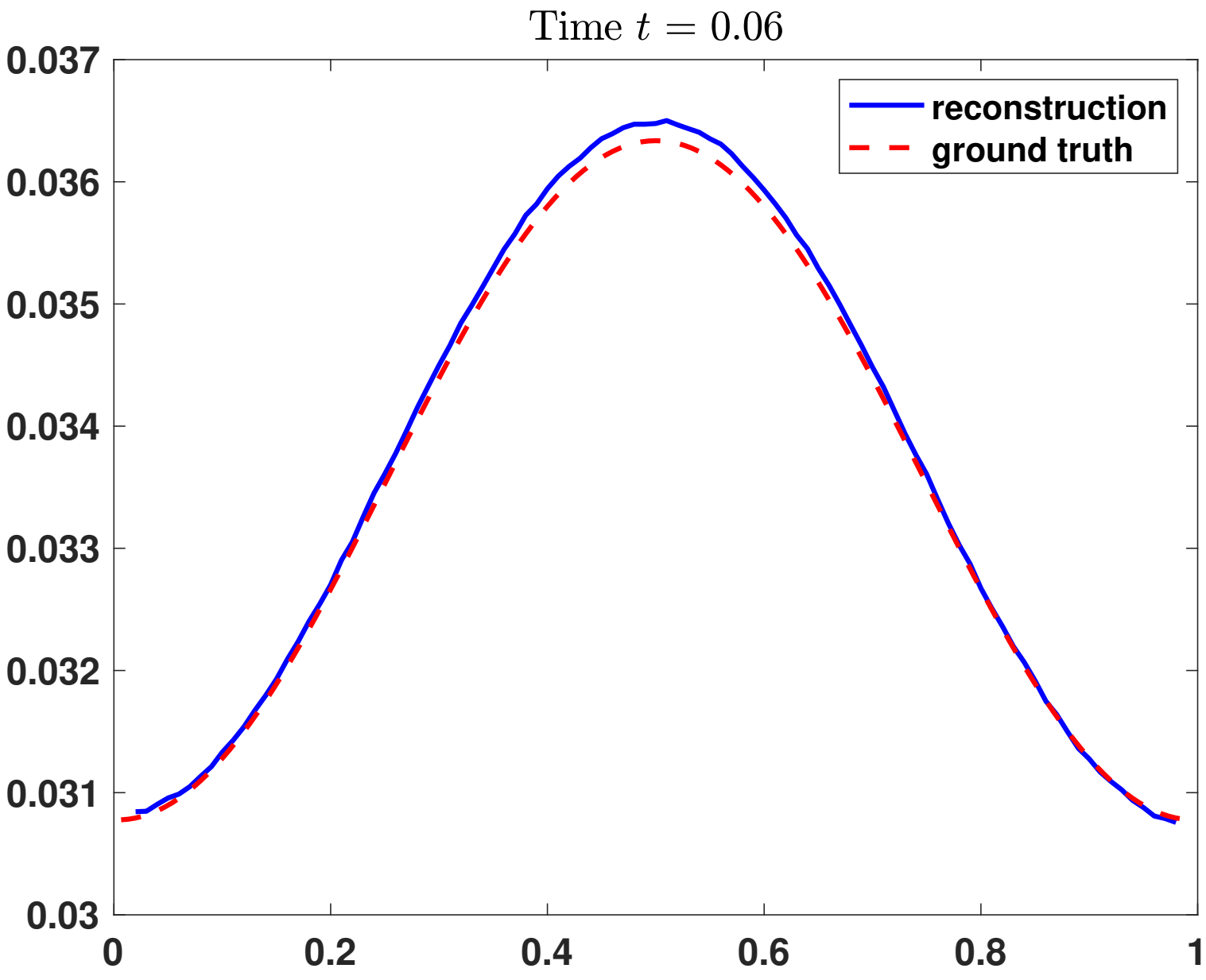
Ensemble Kalman filter ($N = 10^6$)



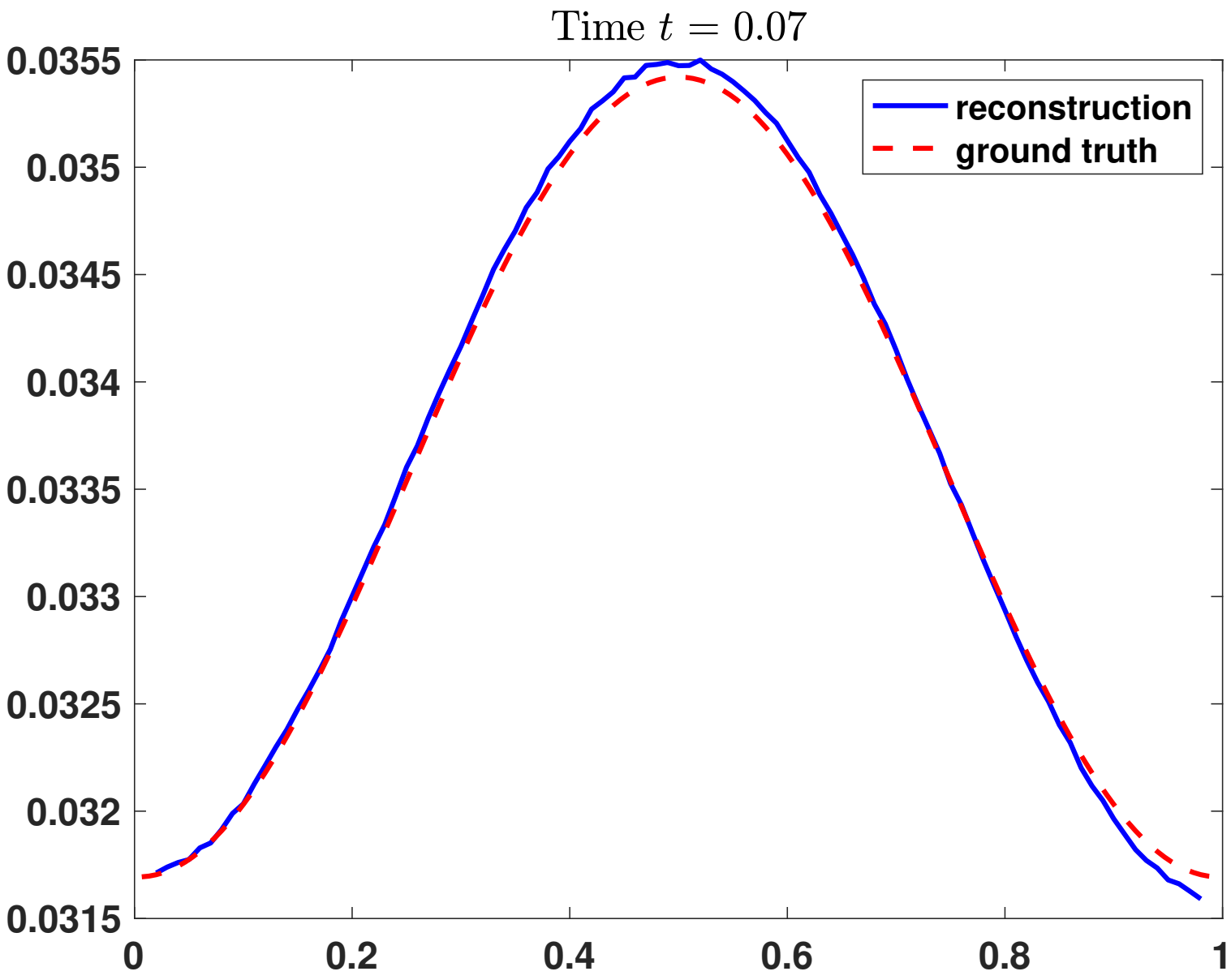
Ensemble Kalman filter ($N = 10^6$)



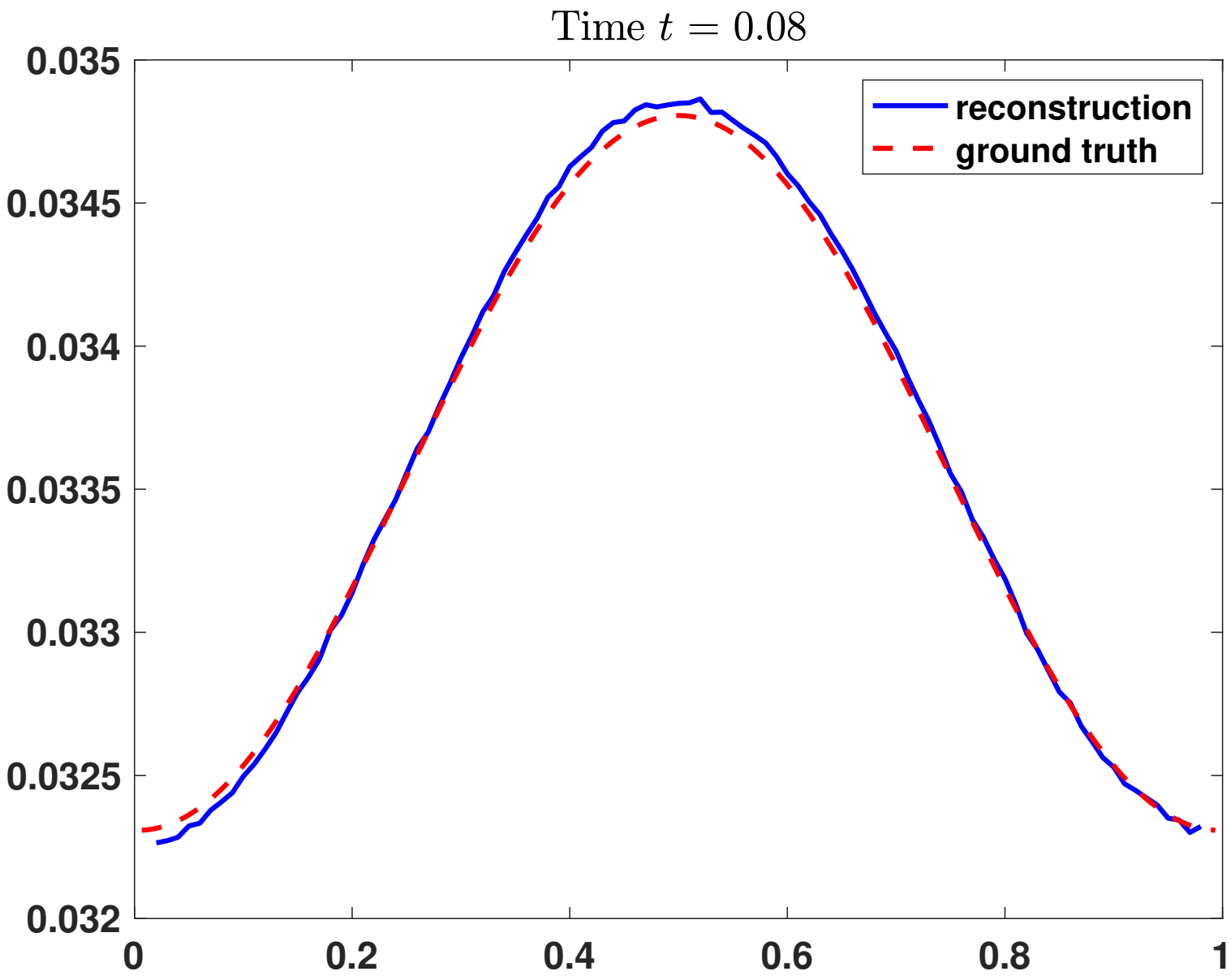
Ensemble Kalman filter ($N = 10^6$)



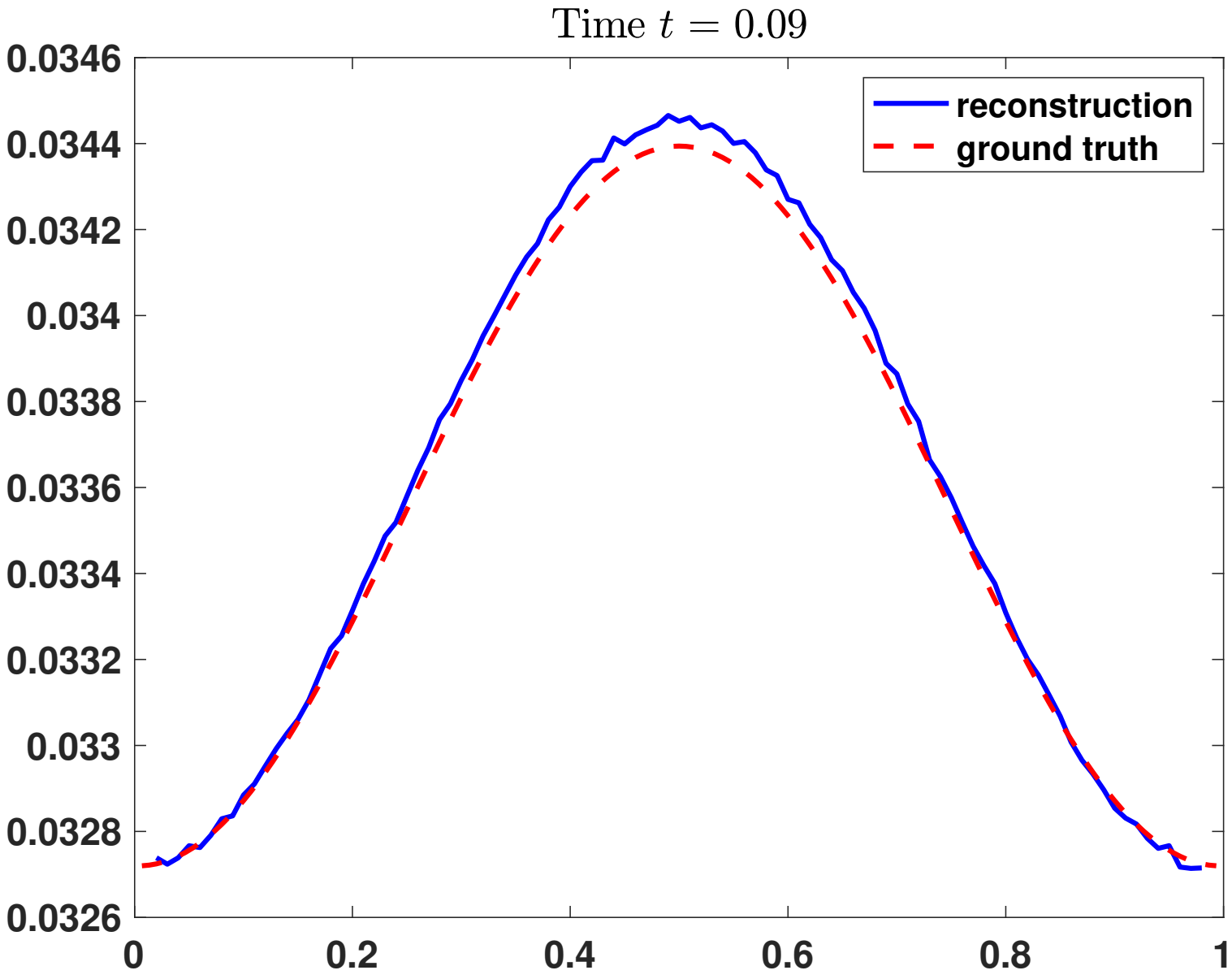
Ensemble Kalman filter ($N = 10^6$)



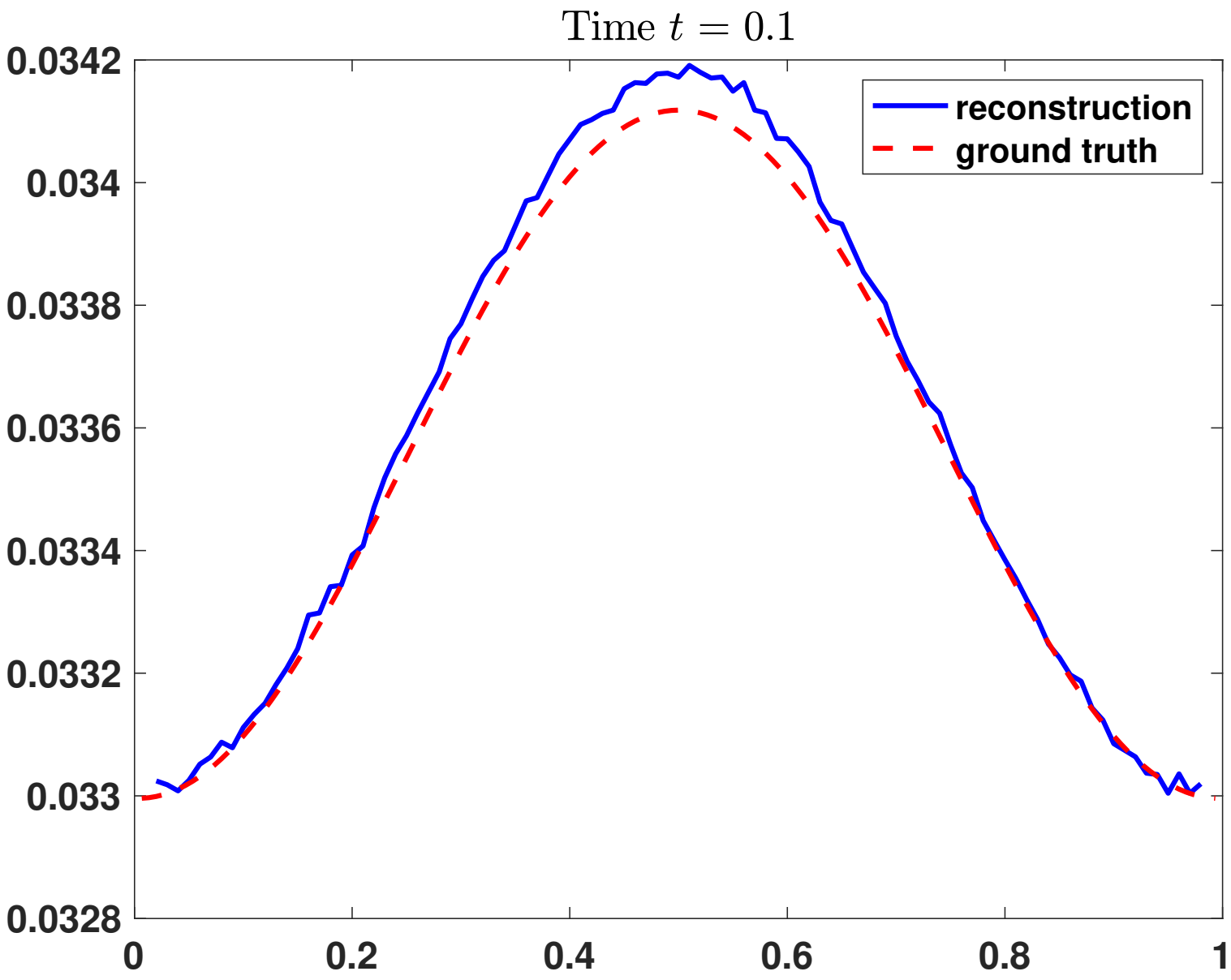
Ensemble Kalman filter ($N = 10^6$)



Ensemble Kalman filter ($N = 10^6$)



Ensemble Kalman filter ($N = 10^6$)



Remarks:

- The Kalman filter is optimal in the sense that it gives the best estimator of the mean in an online setting. (See Theorem 8.6 in Sanz-Alonso, Stuart, Taeb 2018).
- In the linear case ($G(\cdot) = M\cdot$), the ensemble Kalman filter converges to the Kalman filter. When applicable, the ensemble Kalman filter is much more efficient than particle filters. A primary advantage of ensemble methods is that they can provide good state estimation even when the number of particles is *not* large.