Uncertainty quantification for stellar population-kinematics models of galaxies

Progress report June

Fabian Parzer

June 9, 2021

Abstract

In the last meeting, different experiments were proposed to become more familiar with the properties of the problem and our particular approach. Here is a protocol of some of them.

Contents

1 General setup

Let us recap the current approach:

We have the observation operator $\boldsymbol{G} = \boldsymbol{G}(\boldsymbol{f}, \boldsymbol{\theta}_v)$. At the moment, I perform all experiments by computing an exact measurement $y^* = \boldsymbol{G}(\boldsymbol{f}^*, \boldsymbol{\theta}_v^*)$ and then simulating a noisy measurement by adding Gaussian noise:

$$\boldsymbol{y} = \boldsymbol{y}^* + \delta\boldsymbol{\xi}, \qquad \xi \sim \mathcal{N}(0, \boldsymbol{I}),$$

Here, $\delta > 0$ is chosen in such a way that we achieve a prescribed signal-to-noise ratio. The signal-to-noise ratio is defined as

$$\mathrm{snr} := \frac{\|\boldsymbol{y}^*\|}{\|\boldsymbol{y} - \boldsymbol{y}^*\|},$$

and so if we prescribe a signal-to-noise ratio, the noise level δ is determined by

$$\delta = \frac{\|\boldsymbol{y}^*\|}{\mathrm{snr} \cdot \|\boldsymbol{\xi}\|}.$$

Then we compute estimates $\hat{\boldsymbol{f}}^{\mathrm{MAP}}$ and $\hat{\boldsymbol{\theta}}_v^{\mathrm{MAP}}$ by solving the constrained optimization problem

$$\begin{aligned}(\hat{\boldsymbol{f}}^{\mathrm{MAP}}, \hat{\boldsymbol{\theta}}_v^{\mathrm{MAP}}) = \underset{\boldsymbol{f} \in \mathbb{R}^{n_1}, \boldsymbol{\theta}_v \in \mathbb{R}^{n_2}}{\operatorname{argmin}} & \left(\frac{1}{\delta^2} \|\boldsymbol{y} - \boldsymbol{G}(\boldsymbol{f}, \boldsymbol{\theta}_v)\|_2^2 + \beta_1 \left\| \boldsymbol{\Sigma}_1^{-1/2} \boldsymbol{f} \right\|_2^2 + \beta_2 \left\| \boldsymbol{\Sigma}_2^{-1/2} (\boldsymbol{\theta}_v - \bar{\boldsymbol{\theta}}_v) \right\|_2^2 \right), \\ \text{such that} \quad & \boldsymbol{f} \geq \boldsymbol{0}. \end{aligned} \tag{1.1}$$

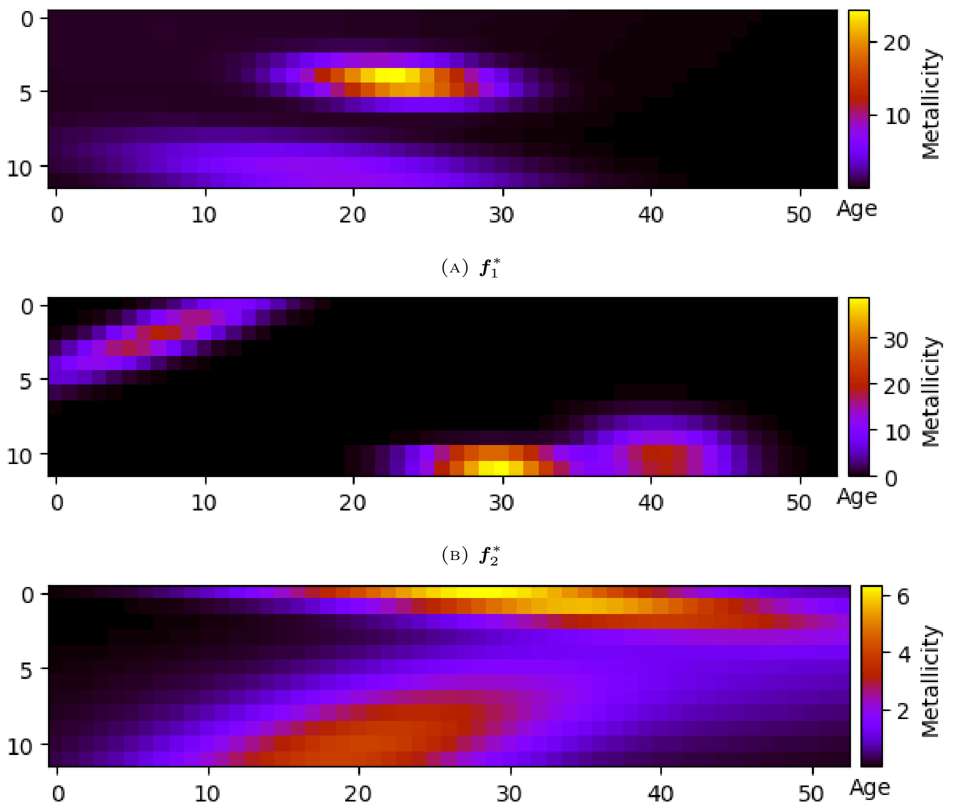
We recall the meaning of the parameters:

- $\boldsymbol{f}$ is the distribution function. In our case, $\boldsymbol{f}$ is an image of shape $(12, 53)$, so $n_1 = 12 \cdot 53 = 636$.
- $\boldsymbol{\theta}_v$ is the hyperparameter for the Gauss-Hermite expansion. It is given by $\boldsymbol{\theta}_v = [V, \sigma, h_0, \ldots, h_m]$, so that it has dimension $n_2 = m + 3$. In our case, $m = 4$, and so $n_2 = 7$.
- $\beta_1 > 0$ and $\beta_2 > 0$ are the regularization parameters for $\boldsymbol{f}$ and $\boldsymbol{\theta}_v$. In all experiments, we used the default values $\beta_1 = 1$ and $\beta_2 = 1$ if not stated otherwise
- $\boldsymbol{\Sigma}_1 \in \mathbb{R}^{n_1 \times n_1}$ is the covariance matrix for $\boldsymbol{f}$. Up until now we have mostly considered the case $\boldsymbol{\Sigma}_1 = \boldsymbol{I}$, but we will see below that a different choice might be more desirable.
- $\bar{\boldsymbol{\theta}}_v = [\bar{V}, \bar{\sigma}, \ldots, \bar{h}_m]$ is our best guess (or prior mean) for $\boldsymbol{\theta}_v$, and $\boldsymbol{\Sigma}_2 \in \mathbb{R}^{n_2 \times n_2}$ is the corresponding covariance matrix. We will usually take it to be a diagonal matrix, i.e.

$$\boldsymbol{\Sigma}_2 = \begin{bmatrix} s_V^2 & 0 & \ldots & 0 \\ 0 & s_\sigma^2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & s_{h_m}^2 \end{bmatrix}$$

  where $s_V, s_\sigma, \ldots$ quantify how close we expect our guesses $\bar{V}, \bar{\sigma}, \ldots$ to be to the truth.

For better comparability of the experiments, I simulated 3 distribution functions $\boldsymbol{f}_1^*$, $\boldsymbol{f}_2^*$ and $\boldsymbol{f}_3^*$ using Prashin's random-Gaussian-mixture simulator (see Figure 1). These were then used as test cases for all the numberical experiments.



(A) $\boldsymbol{f}_1^*$

(B) $\boldsymbol{f}_2^*$

(C) $\boldsymbol{f}_3^*$

FIGURE 1. *The 3 simulated distribution functions.*

For the true value of $\boldsymbol{\theta}_v$, I always used

$$\boldsymbol{\theta}_v^* = \begin{bmatrix} V \\ \sigma \\ h_0 \\ h_1 \\ h_2 \\ h_3 \\ h_4 \end{bmatrix} = \begin{bmatrix} 30 \\ 100 \\ 1 \\ 0 \\ 0 \\ -0.05 \\ 0.1 \end{bmatrix}.$$

In the following experiments, we will use some quantitive criteria to assess the quality of the MAP estimate $(\hat{\boldsymbol{f}}^{\mathrm{MAP}}, \hat{\boldsymbol{\theta}}_v^{\mathrm{MAP}})$. One of those criteria is the **relative data misfit**

$$\mathrm{rdm} := \frac{\left\| \boldsymbol{y} - \boldsymbol{G}(\hat{\boldsymbol{f}}^{\mathrm{MAP}}, \hat{\boldsymbol{\theta}}_v^{\mathrm{MAP}} \right\|)}{\|\boldsymbol{y} - \boldsymbol{y}^*\|}. \tag{1.2}$$

A value close to 1 indicates that the estimate $(\hat{\boldsymbol{f}}^{\mathrm{MAP}}, \hat{\boldsymbol{\theta}}_v^{\mathrm{MAP}})$ is as consistent with the noisy measurement as the true value $(\boldsymbol{f}^*, \boldsymbol{\theta}_v^*)$. A relative data misfit less than 1 can be interpreted as an indicator of overfitting.

We will furthermore also use the **relative reconstruction error**

$$e_f^{\mathrm{rel}} := \frac{\left\| \hat{\boldsymbol{f}}^{\mathrm{MAP}} - \boldsymbol{f}^* \right\|}{\|\boldsymbol{f}^*\|}, \qquad e_{\boldsymbol{\theta}_v}^{\mathrm{rel}} := \frac{\left\| \hat{\boldsymbol{\theta}}_v^{\mathrm{MAP}} - \boldsymbol{\theta}_v^* \right\|}{\|\boldsymbol{\theta}_v^*\|}, \tag{1.3}$$

as a general measure of fit between our estimate $(\hat{\boldsymbol{f}}^{\mathrm{MAP}}, \hat{\boldsymbol{\theta}}_v^{\mathrm{MAP}})$ and the truth.

Finally, we will also use the cost function of the optimization problem Equation 1.1,

$$\phi(\boldsymbol{f}, \boldsymbol{\theta}_v) := \frac{1}{\delta^2} \|\boldsymbol{y} - \boldsymbol{G}(\boldsymbol{f}, \boldsymbol{\theta}_v)\|_2^2 + \beta_1 \left\| \boldsymbol{\Sigma}_1^{-1/2} \boldsymbol{f} \right\|_2^2 + \beta_2 \left\| \boldsymbol{\Sigma}_2^{-1/2} (\boldsymbol{\theta}_v - \bar{\boldsymbol{\theta}}_v) \right\|_2^2, \tag{1.4}$$

as a tool for our analysis. Ideally, the true value $(\boldsymbol{f}^*, \boldsymbol{\theta}_v^*)$ is a global minimum of ϕ, and the value $\phi^{\mathrm{MAP}} = \phi(\hat{\boldsymbol{f}}^{\mathrm{MAP}}, \hat{\boldsymbol{\theta}}_v^{\mathrm{MAP}})$ of the cost function at the MAP estimate is close to $\phi^* = \phi(\boldsymbol{f}^*, \boldsymbol{\theta}_v^*)$. If the reconstruction error of our estimate is large, but ϕ^{MAP} is much smaller than ϕ^*, then this indicates that solving the optimization problem (1.1) does not lead to a good reconstruction and we have to change the cost function, for example by using different values for the regularization parameters or different covariance matrices. If ϕ^{MAP} is much larger than ϕ^*, this indicates that our optimization method did not properly converge, or that ϕ has many local minima.

2 MAP estimation

2.1 Experiment 1: Different regularization operators

So far, all experiments were performed under the choice $\boldsymbol{\Sigma}_1 = \boldsymbol{I}$. But this choice might not be ideal, as it penalizes only the size of each individual pixel, and does not take the smooth structure of the distribution function $\boldsymbol{f}$ into account. Hence we tested other regularization operators aswell.

2.1.1 Ornstein-Uhlenbeck covariance

The Ornstein-Uhlenbeck covariance matrix $\boldsymbol{\Sigma}^{\mathrm{OU}}$ for an image of shape (n_x, n_y) is defined as

$$\boldsymbol{\Sigma}_{i,j}^{\mathrm{OU}} = e^{-\frac{\|\boldsymbol{p}_i - \boldsymbol{p}_j\|}{h}}, \quad i, j = 1, \dots, n_x \cdot n_y, \tag{2.1}$$

where $\boldsymbol{p}_i \in \mathbb{R}^2$ is the normalized position of the i-th pixel, given by

$$\boldsymbol{p}_i = \begin{bmatrix} (i \bmod n_y)/(n_y - 1) \\ \lfloor i/y \rfloor/(n_x - 1) \end{bmatrix},$$

and $h > 0$ is a tunable parameter.

The Ornstein-Uhlenbeck covariance corresponds to the assumption that close pixels are correlated, but the correlation decreases exponentially with distance. The parameter h determines the typical "correlation length" of each pixel. If we use $\boldsymbol{\Sigma}^{\mathrm{OU}}$ instead of the identity as covariance matrix for $\boldsymbol{f}$, it will lead to smoother reconstructions, as drastic changes in value between close pixels are penalized.

2.1.2 Discrete gradient

The discrete gradient of an image $\boldsymbol{F} \in \mathbb{R}^{n_x \times n_y}$ is given as a map $\nabla_{\mathrm{d}} : \mathbb{R}^{n_x \times n_y} \to \mathbb{R}^{n_x \times n_y \times 2}$ with

$$(\nabla_{\mathrm{d}} \boldsymbol{F})_{i,j} = \begin{bmatrix} F_{i+1,j} - F_{i,j} \\ F_{i,j+1} - F_{i,j} \end{bmatrix}, \quad i = 1, \dots, n_x; \quad j = 1, \dots, n_y,$$

where the image is padded with zeros at the boundaries so that the above is well-defined for all i, j. If we flatten the image $\boldsymbol{F}$ to a vector $\boldsymbol{f} \in \mathbb{R}^{n_x \cdot n_y}$, then the discrete gradient operator can be represented by a matrix $\boldsymbol{D} \in \mathbb{R}^{(2n_x \cdot n_y) \times (n_x \cdot n_y)}$.

Using the discrete gradient as regularization operator corresponds to the covariance matrix

$$\boldsymbol{\Sigma}^{\mathrm{DG}} = (\boldsymbol{D}\boldsymbol{D}^\top)^{-1}.$$

If we use this covariance matrix instead of the identity, we penalize not the size of the pixels but the amount of change between neighbouring pixels. Consequently, the reconstruction will be closer to piecewise constant.

2.1.3 Discrete Laplacian

The discrete Laplacian $\Delta_d : \mathbb{R}^{n_x \times n_y} \to \mathbb{R}^{n_x \times n_y}$ of an image $\boldsymbol{F} \in \mathbb{R}^{n_x \times n_y}$ is given by

$$(\Delta_d \boldsymbol{F})_{i,j} = F_{i+1,j} + F_{i,j+1} + F_{i-1,j} + F_{i,j-1} - 4F_{i,j}, \qquad i = 1, \dots, n_x; \quad j = 1, \dots, n_y,$$

where we use reflection at the boundaries to ensure that the above is well-defined. As before, if we flatten the image we obtain a matrix representation of the discrete Laplacian. This leads to a matrix

$\boldsymbol{L} \in \mathbb{R}^{(n_x \cdot n_y) \times (n_x \cdot n_y)}$. As above, we can then penalize the discrete Laplacian of $\boldsymbol{f}$ if we use the covariance matrix

$$\boldsymbol{\Sigma}^{\mathrm{DL}} = (\boldsymbol{L}\boldsymbol{L}^\top)^{-1}.$$

In constrast to the discrete gradient, this choice of covariance matrix does not penalize all change between neighbouring pixels. Instead, it only penalizes nonlinear change, i.e. curvature. Consequently, the reconstruction will be closer to piecewise linear.

2.1.4 Results

We solved (1.1) for the four choices of regularization operators

$$\Sigma_1 \in \{\boldsymbol{I}, \boldsymbol{\Sigma}^{\mathrm{OU}}, \boldsymbol{\Sigma}^{\mathrm{DG}}, \boldsymbol{\Sigma}^{\mathrm{DL}}\}.$$

(for the Ornstein-Uhlenbeck covariance, we used the correlation length $h = 0.1$) In order to evaluate only the regularization on $\boldsymbol{f}$, I fixed $\boldsymbol{\theta}_v$ at its true value $\boldsymbol{\theta}_v^*$, so that (1.1) reduces to a quadratic optimization problem. I always used the default regularization parameter $\beta_1 = 1$. I repeated the experiment for the large-noise (snr = 200) and the low-noise (snr = 2000) case. The results are given in the tables 1-6 and you can find plots of the reconstructions in Appendix A. We make the following observations:

- In all cases, the identity performs worst. Hence, it should probably not be used as a covariance matrix.
- In the low-noise regime (snr = 2000), the Ornstein-Uhlenbeck covariance performs best.
- In the high-noise regime (snr = 200), the situation is less clear. Sometimes the discrete Laplacian performs better, sometimes the Ornstein-Uhlenbeck covariance.
- In any case, the discrete Laplacian performs consistently better than the discrete gradient. Maybe an even higher order differential operator could do better?

	regop	MAP-cost	Truth-cost	rdm	erel f
	Identity	1.05	1.21	1.05	0.9
	Ornstein-Uhlenbeck	0.66	0.74	1.01	0.52
	Discrete-Gradient	0.72	0.98	1.04	0.77
	Discrete-Laplacian	0.67	1.07	1.04	0.75

TABLE 1. *Results of experiment 1 for f_1, snr = 2000. The table shows for every used regularization operator the number of iterations to solve Equation 1.1, the value ϕ^{MAP} of the cost function at the MAP estimate, the value ϕ^* of the cost function at the true parameter value, the relative data misfit (see Equation 1.2) and the relative reconstruction error of $\boldsymbol{f}$ (see Equation 1.3).*

	regop	MAP-cost	Truth-cost	rdm	erel f
	Identity	1.31	1.62	1.15	1.23
	Ornstein-Uhlenbeck	0.75	0.95	1.03	0.71
	Discrete-Gradient	0.88	1.21	1.06	0.87
	Discrete-Laplacian	0.72	1.28	1.05	0.79

TABLE 2. *Results of experiment 1 for f_2, snr = 2000.*

	regop	MAP-cost	Truth-cost	rdm	erel f
	Identity	0.82	0.86	1.01	0.44
	Ornstein-Uhlenbeck	0.58	0.59	1	0.18
	Discrete-Gradient	0.68	0.72	1	0.29
	Discrete-Laplacian	0.56	0.65	1	0.28

TABLE 3. *Results of experiment 1 for f_3, snr = 2000.*

	regop	MAP-cost	Truth-cost	rdm	erel f
0	Identity	0.94	1.21	1.04	1.39
1	Ornstein-Uhlenbeck	0.59	0.73	1	1.07
2	Discrete-Gradient	0.59	0.99	1.01	1.09
3	Discrete-Laplacian	0.54	1.06	1	0.98

Table 4. *Results of experiment 1 for f_1, snr* $= 200$.

	regop	MAP-cost	Truth-cost	rdm	erel f
	Identity	1	1.61	1.08	2.38
	Ornstein-Uhlenbeck	0.64	0.96	1.01	1.49
	Discrete-Gradient	0.7	1.2	1.02	1.49
	Discrete-Laplacian	0.57	1.28	1	1.34

Table 5. *Results of experiment 1 for f_2, snr* $= 200$.

	regop	MAP-cost	Truth-cost	rdm	erel f
	Identity	0.8	0.87	1.01	0.68
	Ornstein-Uhlenbeck	0.56	0.58	1	0.38
	Discrete-Gradient	0.64	0.73	1.01	0.59
	Discrete-Laplacian	0.53	0.64	1	0.42

Table 6. *Results of experiment 1 for f_3, snr* $= 200$.

2.2 Experiment 2: How good does our initial guess for θ_v need to be?

In our last meeting, we noted that the optimization solver is runs into problems when the initial guess $\bar{\boldsymbol{\theta}}_v$ is too far away from $\boldsymbol{\theta}_v^*$, resulting in a poor estimate $(\hat{\boldsymbol{f}}^{\mathrm{MAP}}, \hat{\boldsymbol{\theta}}_v^{\mathrm{MAP}})$. But what does "too far away" mean precisely? To find this out, I performed tests for initial guesses $\bar{\boldsymbol{\theta}}_v$ that were less and less close to the true value $\boldsymbol{\theta}_v^*$, starting with the case were we fix $\boldsymbol{\theta}_v = \boldsymbol{\theta}_v^*$ (i.e. $\bar{\boldsymbol{\theta}}_v = \boldsymbol{\theta}_v^*$ and $\beta_2 = \infty$) and only optimize over $\boldsymbol{f}$ (which leads to a quadratic problem).

The experiments were performed as follows, given a percent in deviation $q \geq 0$.

(i) Create a "bad" starting value by adding noise to the true value of $\boldsymbol{\theta}_v$:

$$\boldsymbol{\theta}_v^q = \boldsymbol{\theta}_v^* + q \cdot \boldsymbol{w},$$

where $\boldsymbol{w} \in \mathbb{R}^7$ is Gaussian noise with covariance $\boldsymbol{\Sigma}_{\boldsymbol{w}} = \mathrm{diag}(30^2, 100^2, 1, 1, 1, 1, 1)$.

(ii) I set $\bar{\boldsymbol{\theta}}_v = \boldsymbol{\theta}_v^q$ and the covariance matrix $\boldsymbol{\Sigma}_2$ equal to the covariance of the noise $q \cdot \boldsymbol{w}$, i.e.

$$\boldsymbol{\Sigma}_2 = q \cdot \mathrm{diag}(30^2, 100^2, 1, 1, 1, 1, 1) \in \mathbb{R}^{7\times 7}.$$

(iii) As explained in Section 1, I simulated a noisy measurement $\boldsymbol{y}^*$ with snr $= 200$ from the true parameter values $\boldsymbol{f}^*$ and $\boldsymbol{\theta}_v^*$.

(iv) Then I solved (1.1) to obtain MAP estimators $\hat{\boldsymbol{f}}^{\mathrm{MAP}} = \hat{\boldsymbol{f}}^{\mathrm{MAP}}(q)$ and $\hat{\boldsymbol{\theta}}_v^{\mathrm{MAP}} = \hat{\boldsymbol{\theta}}_v^{\mathrm{MAP}}(q)$ in dependence on the initial uncertainty q. As prior covariance for $\boldsymbol{f}$, I used the **Ornstein-Uhlenbeck covariance matrix** (see subsubsection 2.1.1).

The results of the numerical experiments are given in Table 7.

	q	niter	MAP-cost	truth-cost	rdm	erel f	erel theta
f1							
	0	2	0.61	0.75	1	1.08	0
	0.003	3	0.6	0.74	1	1.1	0
	0.01	5	0.6	0.74	1	1.09	0.01
	0.03	18	0.6	0.74	1.01	1.22	0.05
	0.05	25	0.57	0.73	1	1.51	0.02
	0.1	42	0.63	0.74	1.07	1.81	0.06
	0.2	50	0.66	0.75	1.09	2.31	0.07
	0.5	45	0.72	0.72	1.19	4.72	0.19
f2							
	0	2	0.64	0.96	1.01	1.51	0
	0.003	3	0.64	0.96	1.01	1.49	0
	0.01	11	0.64	0.97	1.01	1.59	0.01
	0.03	3	0.7	0.96	1.06	1.38	0.02
	0.05	29	0.67	0.95	1.07	1.98	0.04
	0.1	81	0.88	0.97	1.24	2.09	0.04
	0.2	65	0.95	0.96	1.32	3.52	0.11
	0.5	32	2.61	0.96	2.27	6	0.21
f3							
	0	2	0.56	0.57	1	0.36	0
	0.003	3	0.57	0.59	1	0.41	0
	0.01	4	0.57	0.59	1	0.36	0.01
	0.03	19	0.57	0.6	1	0.47	0.02
	0.05	29	0.58	0.59	1.01	0.44	0.05
	0.1	40	0.65	0.58	1.09	0.56	0.05
	0.2	39	0.56	0.59	1.01	1.27	0.03
	0.5	15	2.71	0.59	2.28	1	2.08

TABLE 7. *Results of experiment 2.*

We observe the following:

- The reconstruction is sensitive to the initial guess for $\boldsymbol{\theta}_v$. For values of q larger than 0.01, the reconstruction errors for $\boldsymbol{f}$ and $\boldsymbol{\theta}_v$ are consistently increasing with q.
- For large values of q (sometimes already for $q = 0.05$), the optimization solver struggles a lot, and I had to carefully fine-tune some solver options to achieve convergence. In general, the nonlinear optimization problem Equation 1.1 seems to be quite challenging.

As a conclusion, we probably need a good initial guess for $\boldsymbol{\theta}_v$ ("good" meaning it shouldn't deviate more than 5% from the true value, in the sense specified above) if we want good reconstructions.

2.3 Experiment 3: Different values of β_2

Next, I wanted to look how sensitive the reconstruction method is regarding the regularization parameter β_2. To this end, I repeated the numerical tests for different values of β_2, with 5%-error in the initial guess of $\boldsymbol{\theta}_v$ and a signal-to-noise ratio of 200. The results of the experiment are given in table Table 8. We observe the following:

- The reconstruction is surprisingly insensitive to high values of β_2. The results for values of β_2 between 1 and 100 000 seem to be comparable. In particular, it surprises me that he case where β_2 is extremely large ($\beta_2 = 100000$) performs best in 2 out of 3 cases. Using such a high value for β_2 essentially just corresponds to fixing $\boldsymbol{\theta}_v$ at $\bar{\boldsymbol{\theta}}_v$ and then treating the problem as linear.
- On the other hand, letting $\boldsymbol{\theta}_v$ vary more (using low values of β_2 actually worsens the results. For me, this is counterintuitive: Since $\bar{\boldsymbol{\theta}}_v$ is a noisy guess, letting $\boldsymbol{\theta}_v$ vary should allow the solver to improve the reconstruction by finding values closer to the true value $\boldsymbol{\theta}_v^*$. The solver actually achieves this, in

the sense that it finds an estimate $(\hat{\boldsymbol{f}}^{\mathrm{MAP}}, \hat{\boldsymbol{\theta}}_v^{\mathrm{MAP}})$ at which the cost function attains a smaller value than at the truth. But this does not improve the reconstruction, it worsens it. One explanation could be that the regularization terms are not properly balanced, and the regularization for $\boldsymbol{f}$ implicitly "pushes" the estimate for $\boldsymbol{\theta}_v$ to "bad" values.

Currently, I am rerunning the experiment with a poor initial guess for $\boldsymbol{\theta}_v$ $(q = 0.3)$. For such a poor initial guess, using high values of β_2 should not improve performance, since it corresponds to fixing $\boldsymbol{\theta}_v$ at a bad estimate.

	beta2	niter	MAP-cost	Truth-cost	rdm	erel f	erel theta
f1							
	0.01	102	0.59	0.75	1.01	1.72	0.06
	0.1	50	0.56	0.72	1.01	1.52	0.01
	0.3	23	0.58	0.73	1.01	1.33	0.02
	1	10	0.59	0.74	1.01	1.3	0.06
	3	6	0.59	0.74	1	1.17	0.01
	10	5	0.59	0.74	1	1.23	0.02
	100	2	0.59	0.76	1	1.21	0.01
	1000	2	0.58	0.94	1	1.21	0.04
	10000	2	0.59	1.72	1.01	1.16	0.01
	100000	2	0.61	5.57	1.01	1.12	0.05
f2							
	0.01	118	0.59	0.96	1.01	2.98	0.02
	0.1	38	0.63	0.97	1.02	2.18	0.03
	0.3	14	0.63	0.97	1.01	1.92	0.03
	1	15	0.61	0.95	1.01	2.01	0.06
	3	2	0.63	0.96	1.02	1.77	0.07
	10	5	0.62	0.96	1.01	1.83	0.01
	100	3	0.62	0.98	1.01	1.82	0.02
	1000	3	0.64	1.14	1.01	1.74	0.03
	10000	2	0.65	1.98	1.02	1.72	0.05
	100000	2	0.63	5.9	1.02	1.81	0.04
f3							
	0.01	86	0.57	0.6	1.01	0.71	0.02
	0.1	38	0.56	0.58	1.01	0.65	0.01
	0.3	24	0.56	0.58	1	0.6	0.03
	1	8	0.58	0.6	1.01	0.53	0.01
	3	4	0.57	0.59	1	0.51	0.02
	10	5	0.57	0.61	1	0.53	0.02
	100	3	0.56	0.64	1	0.55	0.03
	1000	3	0.57	1.06	1	0.51	0.03
	10000	2	0.57	2.25	1.01	0.49	0.02
	100000	2	0.57	4.7	1.01	0.47	0.03

Table 8. *Results of experiment 3.*

3 Uncertainty quantification

3.1 Setup

We recall our approach to uncertainty quantification. Given the MAP estimate $(\hat{\boldsymbol{f}}^{\mathrm{MAP}}, \hat{\boldsymbol{\theta}}_v^{\mathrm{MAP}})$, we linearize our model around this estimate. This leads to the statistical model

$$\begin{aligned} \boldsymbol{y}^* &= \boldsymbol{G}(\hat{\boldsymbol{f}}^{\mathrm{MAP}}, \hat{\boldsymbol{\theta}}_v^{\mathrm{MAP}}) + \boldsymbol{G}'(\hat{\boldsymbol{f}}^{\mathrm{MAP}}, \hat{\boldsymbol{\theta}}_v^{\mathrm{MAP}}) \begin{bmatrix} \boldsymbol{f} - \hat{\boldsymbol{f}}^{\mathrm{MAP}} \\ \boldsymbol{\theta}_v - \hat{\boldsymbol{\theta}}_v^{\mathrm{MAP}} \end{bmatrix} + \boldsymbol{w}, \\ \boldsymbol{w} &\sim \mathcal{N}(0, \delta^2 \boldsymbol{I}), \\ \boldsymbol{f} &\sim \mathcal{N}(0, \beta_1^{-1} \boldsymbol{\Sigma}_1), \\ \boldsymbol{\theta}_v &\sim \mathcal{N}(\bar{\boldsymbol{\theta}}_v, \beta_2^{-1} \boldsymbol{\Sigma}_2), \end{aligned} \tag{3.1}$$

with the associated linearized costfunction

$$\begin{aligned} \phi^{\mathrm{lin}}(\boldsymbol{f}, \boldsymbol{\theta}_v) = \frac{1}{2\delta^2} \left\| \boldsymbol{y}^* - \boldsymbol{G}(\hat{\boldsymbol{f}}^{\mathrm{MAP}}, \hat{\boldsymbol{\theta}}_v^{\mathrm{MAP}}) - \boldsymbol{G}'(\hat{\boldsymbol{f}}^{\mathrm{MAP}}, \hat{\boldsymbol{\theta}}_v^{\mathrm{MAP}}) \begin{bmatrix} \boldsymbol{f} - \hat{\boldsymbol{f}}^{\mathrm{MAP}} \\ \boldsymbol{\theta}_v - \hat{\boldsymbol{\theta}}_v^{\mathrm{MAP}} \end{bmatrix} \right\|_2^2 + \beta_1 \left\| \boldsymbol{\Sigma}_1^{-1/2} f \right\|_2^2 \\ + \beta_2 \left\| \boldsymbol{\Sigma}_2^{-1/2} (\boldsymbol{\theta}_v - \bar{\boldsymbol{\theta}}_v) \right\|_2^2 . \end{aligned} \tag{3.2}$$

Let us define the combined parameter vector

$$\boldsymbol{x} := \begin{bmatrix} \boldsymbol{f} \\ \boldsymbol{\theta}_v \end{bmatrix}$$

We then use the concentration of measure approximation to define an approximate **posterior credible region** centered at the MAP estimate:

$$C_\alpha(\boldsymbol{x}) := \left\{ \boldsymbol{x} : \phi^{\mathrm{lin}}(\boldsymbol{x}) \le \phi^{\mathrm{MAP}} + n(\tau_\alpha + 1) \right\}, \tag{3.3}$$

where $\phi^{\mathrm{MAP}} = \phi^{\mathrm{lin}}(\boldsymbol{x}^{\mathrm{MAP}})$, $\tau_\alpha = \sqrt{\frac{16 \log(3/\alpha)}{n}}$, $n = n_1 + n_2$ is the combined dimension of the parameters $\boldsymbol{f}$ and $\boldsymbol{\theta}_v$, and α is a credibility parameter ($\alpha = 0.05$ corresponds to the standard 95%-credibility region).

3.2 Local credible intervals

3.2.1 Recapitulation of the definition

The visualization of the posterior credible $C_\alpha(\boldsymbol{x})$ is a nontrivial task, since it is a set that lives in a very high-dimensional space. Last time, I presented a few possibilities. For example, using sampling from a truncated Gaussian, I tried to compute *projected credible intervals*, which are the theoretically "cleanest" way to visualize the posterior credible region $C_\alpha(\boldsymbol{x})$ in a pixel-by-pixel way. However, this didn't work because the sampler did not converge fast enough, and it was also rather slow.

Therefore, I wanted to try to compute so-called *local credible intervals* next.

We recall the definition of local credible intervals: Suppose we are given the posterior credible region $C_\alpha(\boldsymbol{x})$. Let us say the parameter space $\mathbb{R}^n$ (where $n = \dim(\boldsymbol{f}) + \dim(\boldsymbol{\theta}_v)$) is partitioned into subspaces determined by index sets $\mathcal{I}_1, \ldots, \mathcal{I}_J \subset \{1, \ldots, n\}$ (for the distribution function $\boldsymbol{f}$, you can think of the partition elements as chunks of 2×2 or 3×3 pixels, so called "superpixels". Then, for each partition element $\mathcal{I}_j = \{i_1, \ldots, i_{k_j}\}$, we define the vector $\boldsymbol{x}(\xi) \in \mathbb{R}^{n_1}$ by

$$x(\xi)_i = \begin{cases} x_i^{\mathrm{MAP}} + \xi, & \text{if } i \in \mathcal{I}_j, \\ x_i^{\mathrm{MAP}}, & \text{otherwise}, \end{cases} \qquad i = 1, \ldots, n_1.$$

Outside the partition element $\mathcal{I}_j$, this vector is fixed to the MAP estimate $\boldsymbol{x}^{\mathrm{MAP}}$. Inside $\mathcal{I}_j$, the constant value $\xi \in \mathbb{R}$ is added to each pixel. [1] Next, we find the smallest and largest value of $\xi \in \mathbb{R}$ for which $\boldsymbol{x}(\xi)$

[1] The original definition of the local credible interval in the paper [1] was slightly different: Instead of adding ξ to the MAP estimate, the value of f_j^ξ was set equal to ξ in the superixel $\mathcal{I}_j$. However, in my numerical experiments I noted that this often leads to a situation where $\boldsymbol{x}(\xi) \notin C_\alpha(\boldsymbol{x})$, for any $\xi \in \mathbb{R}$. Hence, I had to modify the definition so that at least $\boldsymbol{x}^0 \in C_\alpha(\boldsymbol{x})$ always holds. As you will see, this modification still did not lead to good results.

still lies in $C_\alpha(\boldsymbol{x})$:

$$\begin{aligned} \xi_{j,-} &:= \min_{\xi \in \mathbb{R}} \{\, \xi : \boldsymbol{x}(\xi) \in C_\alpha(\boldsymbol{x}) \,\} \in \mathbb{R}, \\ \text{and} \quad \xi_{j,+} &:= \max_{\xi \in \mathbb{R}} \{\, \xi : \boldsymbol{x}(\xi) \in C_\alpha(\boldsymbol{x}) \,\} \in \mathbb{R}. \end{aligned}$$

Then, the multidimensional interval (or hypercube)

$$[\boldsymbol{x}_{\mathcal{I}_j}^{\xi_{j,-}}, \boldsymbol{x}_{\mathcal{I}_j}^{\xi_{j,+}}] \subset \mathbb{R}^{k_j},$$

is the local credible interval for the partition element $\mathcal{I}_j$.

3.2.2 Problems

There is one major problem with local credible intervals, as they are defined above, that is not mentioned in the paper [1] and the follow-up paper [2]. For clarity, let us identify the parameter space $\mathbb{R}^n$ with an image domain and the partition elements with superpixels – chunks of 2×2 or 3×3 pixels. In the computation of local credible intervals, we are varying only the pixels inside one superpixel at a time, while keeping the rest of the image fixed. However, if neighbouring superpixels are correlated, then this will underestimate the posterior uncertainty, since it ignores the case where multiple superpixels vary *together*. Of course, one can argue that the correlations in an image are mostly local, so that if we choose the superpixels large enough, they will be roughly independent. But even if this were the case, we would not capture the whole posterior uncertainty, since we only vary the superpixel by a constant value. As an example, it might be unlikely that two neighbouring pixels both are much larger than the MAP estimate *at the same time*, while it might be likely that one of them is larger while at the same time the other one is smaller. To state this more concisely, the way in which local credible intervals are constructed runs into problems when neighbouring pixels are *negatively correlated*. This situation is visualized in Figure 2: Let's say a superpixel $\mathcal{I}_j$ only consists of two pixels, so that we can project the posterior credible region on these two pixels and visualize it as a two-dimensional ellipse. The local credible interval for this superpixel then measures the width of that ellipse, but only in the direction $\pm \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. However, in the pictured case the x_1- and x_2-variable are negatively correlated, so that the local credible interval only covers a small part of the credible region. Note that in this particular example, the true value lies outside the 95%-local credible interval, even though it lies inside the 95%-credible region.

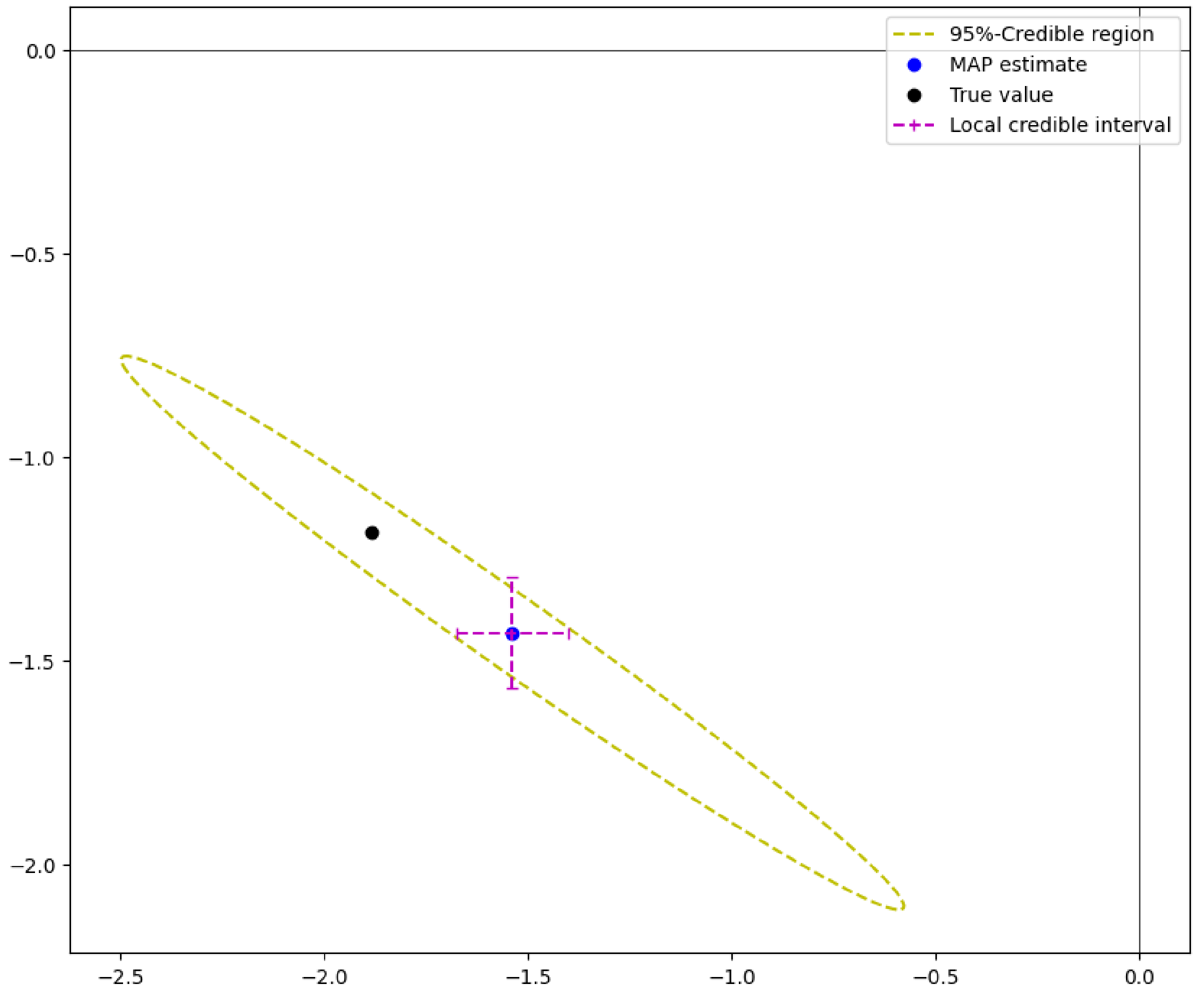


FIGURE 2. *Picture of a credible region and the corresponding local credible intervals.*

3.2.3 Local credible rectangles

We conclude that in some situations, the local credible intervals can drastically underestimate the posterior uncertainty. But there should be a better way. Take another look at Figure 2. The problem with the local credible interval is that it approximates the credible region by fitting a line segment in direction $\begin{bmatrix}1\\1\end{bmatrix}$ inside the ellipse and then builds the square spanned by this line segment. Since the main axis of the ellipse is almost orthogonal to the direction $\begin{bmatrix}1\\1\end{bmatrix}$, this leads to a poor approximation. Instead, a more adaptive approach would be to find the *smallest rectangle that contains the credible region.*

Let us say that the we have again a partition into index sets $\mathcal{I}_1, \dots, \mathcal{I}_J$. Given a partition element $\mathcal{I}_j = \{i_1, \dots, i_{k_j}\}$ of size k_j, we can then fix $\boldsymbol{x}$ outside this partition element and only consider the variation of the components inside $\mathcal{I}_j$. This leads to a k_j-dimensional region

$$C_\alpha(\boldsymbol{x}|\mathcal{I}_j) := \left\{ \boldsymbol{z} \in \mathbb{R}^{k_j} : \boldsymbol{x}_j(\boldsymbol{z}) \in C_\alpha(\boldsymbol{x}) \right\},$$
$$\text{where} \quad x_j(\boldsymbol{z})_i := \begin{cases} z_j, & \text{if there is } i_j \in \mathcal{I}_j \text{ with } i = i_j, \\ x_i^{\mathrm{MAP}}, & \text{otherwise,} \end{cases}$$

which contains all possible values for $\boldsymbol{x}_{\mathcal{I}_j}$ such that $\boldsymbol{x} \in C_\alpha(\boldsymbol{x})$ if we set all components outside $\mathcal{I}_j$ equal to the corresponding components of $\boldsymbol{x}^{\mathrm{MAP}}$.

We can then determine the smallest (hyper-)rectangle that contains $C_\alpha(\boldsymbol{x}|\mathcal{I}_j)$ by solving the optimization problems

$$\begin{aligned} z_l^{j,-} &= \min_{\boldsymbol{z}\in\mathbb{R}^{k_j}} \{ z_l : \boldsymbol{z} \in C_\alpha(\boldsymbol{x}|\mathcal{I}_j) \}, \\ z_l^{j,+} &= \max_{\boldsymbol{z}\in\mathbb{R}^{k_j}} \{ z_l : \boldsymbol{z} \in C_\alpha(\boldsymbol{x}|\mathcal{I}_j) \}, \qquad l = 1,\dots,k_j. \end{aligned} \tag{3.4}$$

The desired rectangle is given by

$$[\boldsymbol{z}^{j,-}, \boldsymbol{z}^{j,+}] \subset \mathbb{R}^{k_j},$$

and we will call it the **j-th local credible rectangle**[2]

Remark 3.1 *Let me explain why we still partition the parameter space: Of course, one could just try to compute the smallest rectangle that contains the whole credible region $C_\alpha(\boldsymbol{x}) \subset \mathbb{R}^n$. However, in that case (3.4) would be an optimization problem in the high-dimensional space $\mathbb{R}^n$, which is computationally expensive. Since we have to solve (3.4) n-times, this would make the computation time very long. By using partition elements with sizes between, say, 1 and 10, we only have to solve n one-to-ten-dimensional optimization problems, which is orders of magnitude faster. Thus the projection on superpixels represents a compromise between exact uncertainty quantification and computational cost. It also counteracts the overestimation error, since in general the set in $\mathbb{R}^n$ spanned by all the local credible rectangles will be smaller than the smallest rectangle that contains the whole credible region $C_\alpha(\boldsymbol{x})$. On the downside, if we project on partition elements, then the local credible rectangles are not a conservative approximation anymore, since the set spanned by those regions will not necessarily contain the whole credible region $C_\alpha(\boldsymbol{x})$ (it will, however, if the partition elements are uncorrelated).*

Once we have computed $\boldsymbol{z}^{j,-}$ and $\boldsymbol{z}^{j,+}$ for all partition elements $\mathcal{I}_1, \dots, \mathcal{I}_J$, we can combine them into two vectors $\boldsymbol{z}^- \in \mathbb{R}^n$ and $\boldsymbol{z}^+ \in \mathbb{R}^n$, where

$$z_k^- = z_{i_k}^{j,-} \text{ and } z_k^+ = z_{i_k}^{j,+} \qquad \text{for all } i_k \in \mathcal{I}_j, \text{ and all } j = 1,\dots,J.$$

These images can then be interpreted as $(1-\alpha)$-credible lower and upper bounds for $\boldsymbol{x}$.

In Figure 3, the local credible interval is compared with the local credible retangle described above. One sees that the rectangle covers the whole credible region, and also contains the true value.

[2] Maybe the name "local credible rectangle" is not informative, since local credible intervals as defined above are also (hyper-)rectangles. But I didn't come up with something better.

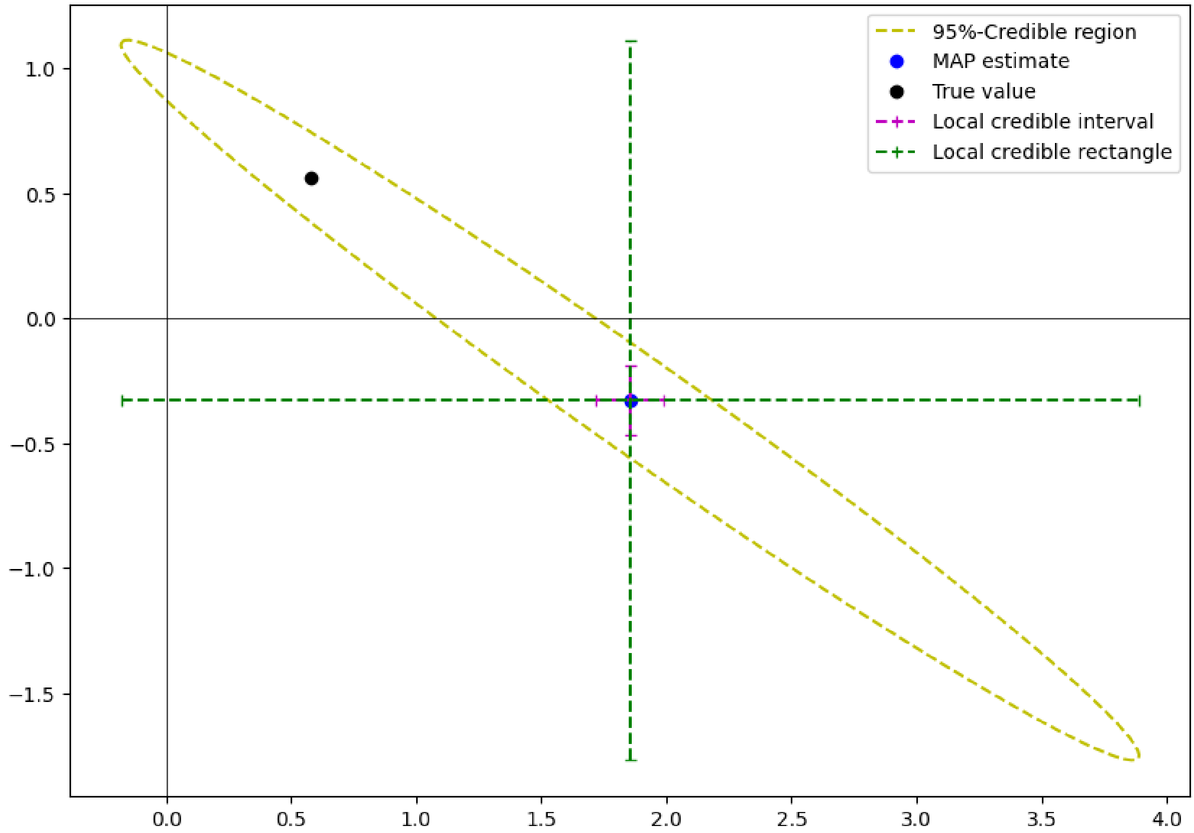
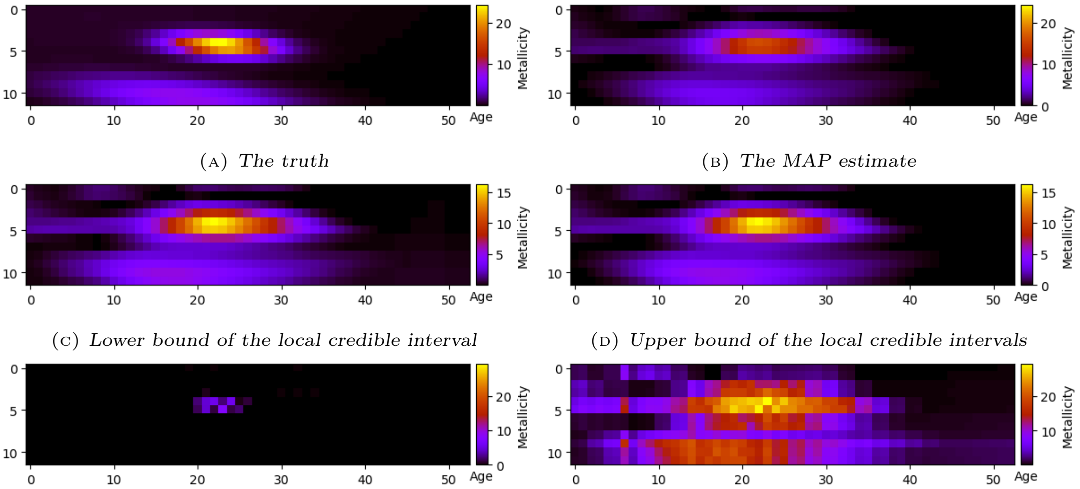


FIGURE 3. *Picture of a credible region, the corresponding local credible intervals, and the local credible rectangles.*

Of course, this is a constructed counterexample. If the x_1- and x_2-variable were uncorrelated, then the credible region would be a circle and the local credible intervals and rectangles would coincide. If they were positively correlated, the eror would be smaller the closer their correlation is to 1. However, it seems that the very phenomenon demonstrated in Figure 3 arises in our problem (3.1). In Figure 4 you can compare the local credible intervals and the local credible rectangles for the distribution function $\boldsymbol{f}$. We observe that the lower and upper bound of the local credible intervals basically coincide, even though the MAP estimate $\hat{\boldsymbol{f}}^{\mathrm{MAP}}$ differs considerable from the true parameter $\boldsymbol{f}^*$. In contrast, the rectangular credible interval indicates a much larger uncertainty (the values of the true distribution function are (with a few exceptions) inside the local credible rectangles). In conclusion, the local credible intervals as proposed in [1] do not seem to capture the posterior uncertainty very well in our case, and hence I suggest we rather use the more conservative local credible rectangles. Nevertheless, it might worthwhile to think more about this problem: What are good visualizations for high-dimensional credible regions?



(A) *The truth*

(B) *The MAP estimate*

(C) *Lower bound of the local credible interval*

(D) *Upper bound of the local credible intervals*

(E) *Lower bound of the rectangular credible intervals*

(F) *Upper bound of the rectangular credible intervals*

FIGURE 4. *Comparing local credible intervals and local credible rectangles*

3.3 Experiment 4: Local credible rectangles for different regularization operators – θ_v fixed

In this experiment, I fixed $\boldsymbol{\theta}_v$ at the true value $\boldsymbol{\theta}_v^*$ so that the forward operator $\boldsymbol{G}$ reduces to a linear operator. I then computed the rectangular credible intervals for the MAP estimators corresponding to the different regularization operators (identity, Ornstein-Uhlenbeck, discrete gradient and discrete Laplacian). I did this both for the small-noise (snr = 2000) and the high-noise (snr = 200) case. As a partition, I used a simple grid of 2 × 2 superpixels.

I observed that the computation of the rectangular local credible regions was pretty fast. It took only 2 minutes to produce the complete uncertainty quantification, which is basically as fast as the computation of local credible intervals, even though the optimization problems are 4-dimensional instead of 1-dimensional ones. But the optimization problems are very simple, and still low-dimensional enough so that the computation time is not significantly larger than the overhead of setting up the problems[3]. Hence, one can say that this approach to uncertainty quantification is computationally efficient, and very scalabe: It scales linearly in the number of superpixels, and cubic in the size of the superpixels.

In order to quantitatively evaluate the fit of the credible intervals, I computed how much of the ground truth lies outside the credible interval. This was done by computing the underestimation error

$$\mathrm{EUQ} := \left\| (\boldsymbol{f}^* - \boldsymbol{\xi}^-)_+ + (\boldsymbol{\xi}^+ - \boldsymbol{f}^*)_+ \right\|_1,$$

where $\boldsymbol{x}_+$ denotes the positive part of the vector $\boldsymbol{x}$, i.e.

$$x_i^+ := \max(x_i, 0), \qquad i = 1, \ldots, n.$$

On the other hand, we also do not want to have credible intervals that are unnecessarily large. For this reason, I also computed a measure for the size of the credible intervals relative to the true deviation $\boldsymbol{f}^* - \hat{\boldsymbol{f}}^{\mathrm{MAP}}$, defined as

$$\mathrm{SUQ} := \frac{1}{n} \sum_{i=1}^{n} \frac{\xi_i^+ - \xi_i^-}{\max(\left| f_i^{\mathrm{MAP}} - f^* \right|, \epsilon)}.$$

(The denominator is capped below at ϵ to avoid division by small numbers. I used the value $\epsilon = 1$). A value of s_{UQ} close to 1 means that the local credible intervals are a good representation of the true distance of the MAP estimate to the ground truth, a value less than 1 indicates underestimation, and a value of more than 1 indicates overestimation of the uncertainty.

[3] I also did some tests with superpixels of size 3 × 3. There, the computations took significantly more time (but still only minutes, not hours)

The results are presented in Table 9. I also plotted some (not all) credible intervals in Appendix B.

	snr	operator	truth in CR	EUQ	SUQ
f1					
	2000	Identity	False	0.01095	4.0
	2000	Ornstein-Uhlenbeck	True	0.00464	4.6
	2000	Gradient	False	0.03542	4.0
	2000	Laplacian	False	0.05406	3.5
	200	Identity	False	0.00949	8.2
	200	Ornstein-Uhlenbeck	True	0.00418	7.0
	200	Gradient	False	0.00556	7.7
	200	Laplacian	False	0.06505	5.5
f2					
	2000	Identity	False	0.04048	3.7
	2000	Ornstein-Uhlenbeck	False	0.02175	3.0
	2000	Gradient	False	0.02053	3.0
	2000	Laplacian	False	0.07124	2.2
	200	Identity	False	0.12783	5.7
	200	Ornstein-Uhlenbeck	False	0.07115	4.8
	200	Gradient	False	0.07313	4.6
	200	Laplacian	False	0.15312	3.6
f3					
	2000	Identity	True	0.00040	5.9
	2000	Ornstein-Uhlenbeck	True	0.00126	6.4
	2000	Gradient	True	0.00070	6.3
	2000	Laplacian	True	0.00024	5.6
	200	Identity	True	0.00000	8.5
	200	Ornstein-Uhlenbeck	True	0.00000	9.3
	200	Gradient	True	0.00000	8.6
	200	Laplacian	True	0.00000	7.4

Table 9. *A quantitative evaluation of the uncertainty quantification for different regularization operators, tested at the 3 test distributions f_1, f_2 and f_3. The third column indicates whether the ground truth lies in the credible region. Note that this does not imply that the ground truth lies in the local credible rectangles.*

We observe the following:

- A larger signal-to-noise ratio (smaller noise) leads to tighter credible intervals and less posterior uncertainty. This is consistent with basic intuition.
- The results confirm the earlier observation that the Ornstein-Uhlenbeck covariance is probably the most appropriate choice. For the Ornstein-Uhlenbeck covariance, the ground truth lies inside the credible region in 4/6 cases, and the underestimation error is usually (except one case) the smallest. But this might also have to do with the fact that the Ornstein-Uhlenbeck covariance leads to rather large credible rectangles.
- The Laplacian always leads to the smallest credible rectangles, but at the cost of a usually larger underestimation error. This could be interpreted as evidence that using the discrete Laplacian as regularization operator leads to overregularization or oversmoothing.
- In general, but especially in the high-noise case, the credible rectangles are large (take a look at the SUQ values, where a value of 1 is the ideal). However, it is unclear whether this is solely due to the ill-posedness of the problem, or also because the approximations we make are too conservative. We could probably reduce the size of the credible rectangles by introducing more constraints or using a stronger form of regularization.
- Most of the times, the credible rectangles contain the zero function. Note that this does not necessarily mean that the zero function is also in the credible region, but that for every pixel there is an image $\boldsymbol{f}$ inside the credible region that is zero at that pixel.

3.4 Experiment 5: Rectangular credible intervals for the nonlinear model

In this experiment, I wanted to see how the uncertainty quantification works for the nonlinear case, that is if we do not fix $\boldsymbol{\theta}_v$. In that case, we also obtain credible intervals for the components $V, \sigma, \ldots, h_m$ of $\boldsymbol{\theta}_v$, which we can then compare to the true values. I performed the experiment with the Ornstein-Uhlenbeck covariance since it yielded the best results in the previous experiments, with a signal-to-noise ratio of 200 and an initial guess for $\boldsymbol{\theta}_v$ perturbed by 3% ($q = 0.03$, see experiment 2). For the regularization parameter, I used the value $\beta_2 = 3$ (see experiment 3).

The computation of the credible rectangles was again very efficient. In the nonlinear case, the computation of the MAP estimates takes many more iterations, so that in this case the computational effort of the uncertainty quantification is small relative to the cost of the MAP estimation. The only problems were encountered when computing the credible rectangles for $\boldsymbol{\theta}_v$, where the optimizer had a surprisingly hard time solving the problem (3.4). This is probably due to ill-conditioning of the linearized model.

The results are given in Table 10. Again, I computed EUQ and SUQ to allow for quantitative comparisons. For $\boldsymbol{\theta}_v$, SUQ was computed with $\epsilon = 0.1$.

Further figures are plotted in Appendix C.

	snr	truth in CR	EUQ (f)	EUQ (theta v)	SUQ (f)	SUQ (theta v)
f1	200	True	1.00	0.0031	7.2	1.2
f2	200	False	1.08	0.1827	4.6	10.7
f3	200	True	1.00	0.1298	9.7	7.5

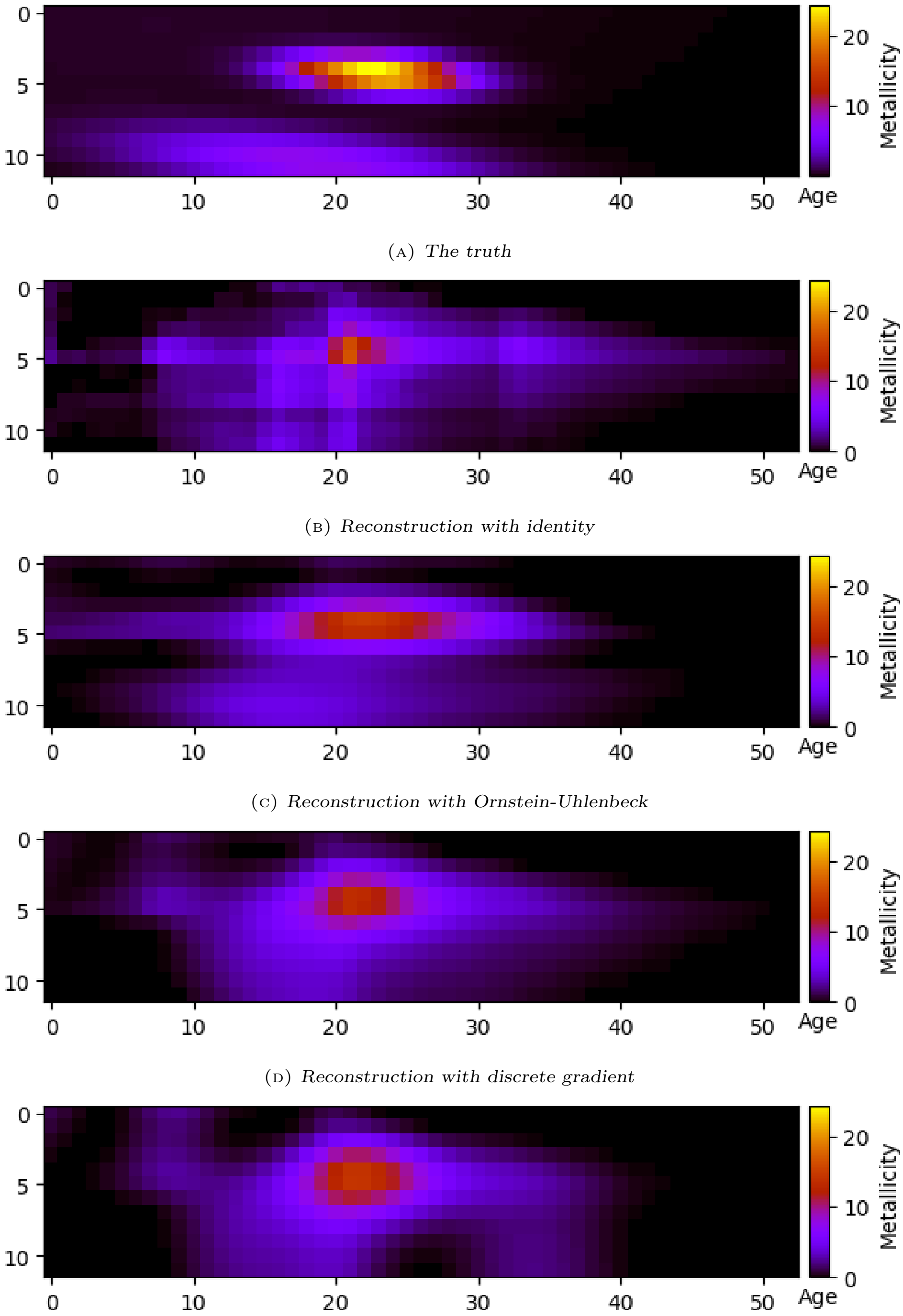
TABLE 10. *The results for experiment 5.*

Remark 3.2 *For the test function f_1, the optimization solver did not converge but became stuck at a point where it could not reduce the cost function. I had to increase the regularization parameter up to a value of $\beta_2 = 100$ until the optimization method converged to a stationary point. This did not affect the results for $\boldsymbol{f}$ very much, but it lead to slightly smaller error bars for $\boldsymbol{\theta}_v$ (see Figure 25).*

We observe the following:

- The error bars for V and σ are sometimes extremely small. This might be due to the linearization. In constrast, the error bars for $h_0, \ldots, h_4$ look reasonably good. In general, the error bars tend to be too large ($SUQ > 1$).
- The reconstructions of the distribution functions look similar as in the case where $\boldsymbol{\theta}_v$ is fixed (compare Figure 23 with Figure 16). However, letting $\boldsymbol{\theta}_v$ vary leads to much larger local credible rectangles.

A Additional figures for experiment 1



(A) *The truth*

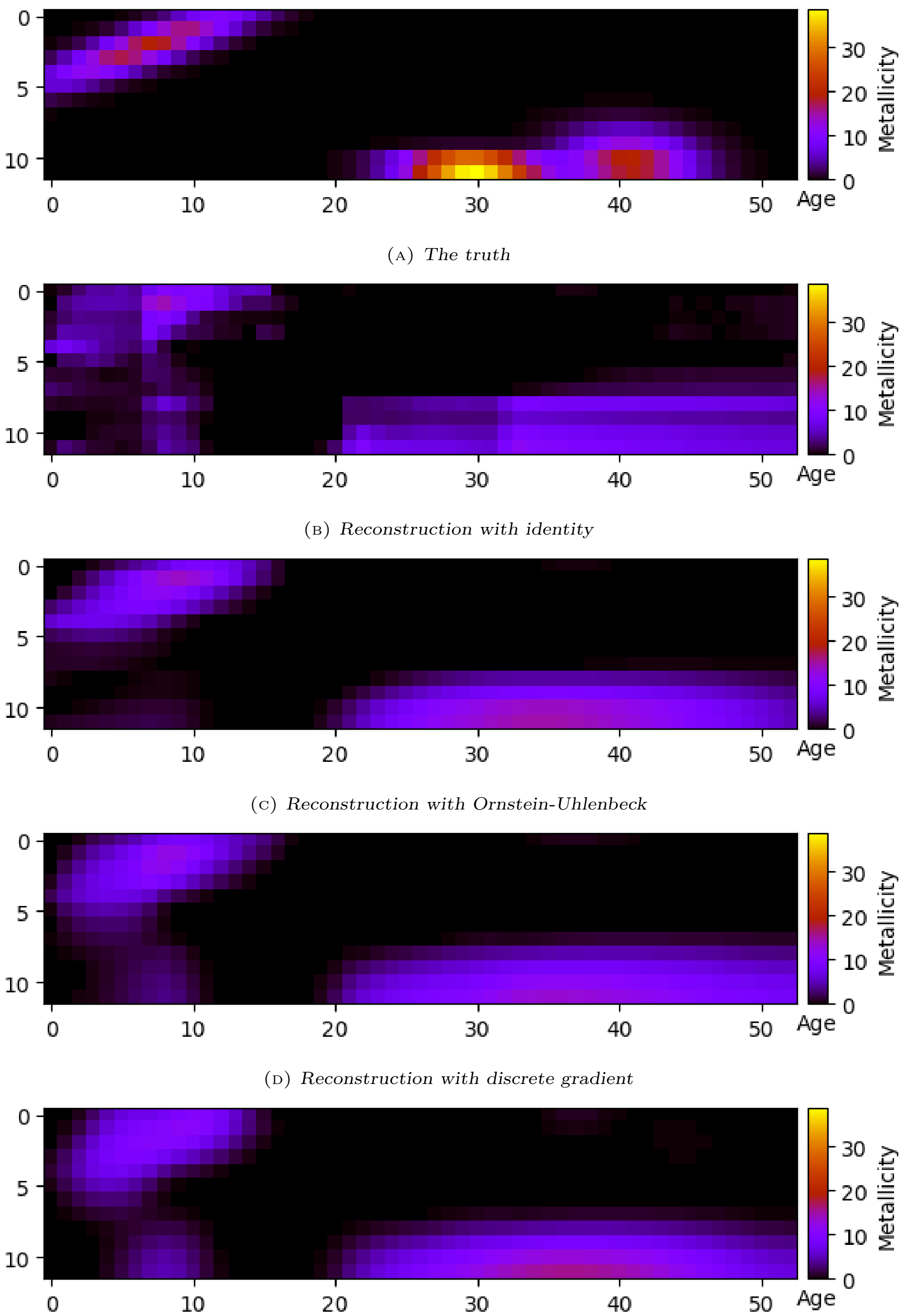
(B) *Reconstruction with identity*

(C) *Reconstruction with Ornstein-Uhlenbeck*

(D) *Reconstruction with discrete gradient*

(E) *Reconstruction with discrete Laplacian*

FIGURE 5. *The results of experiment 1 for f_1 with different regularization operators, snr =* 2000



(A) *The truth*

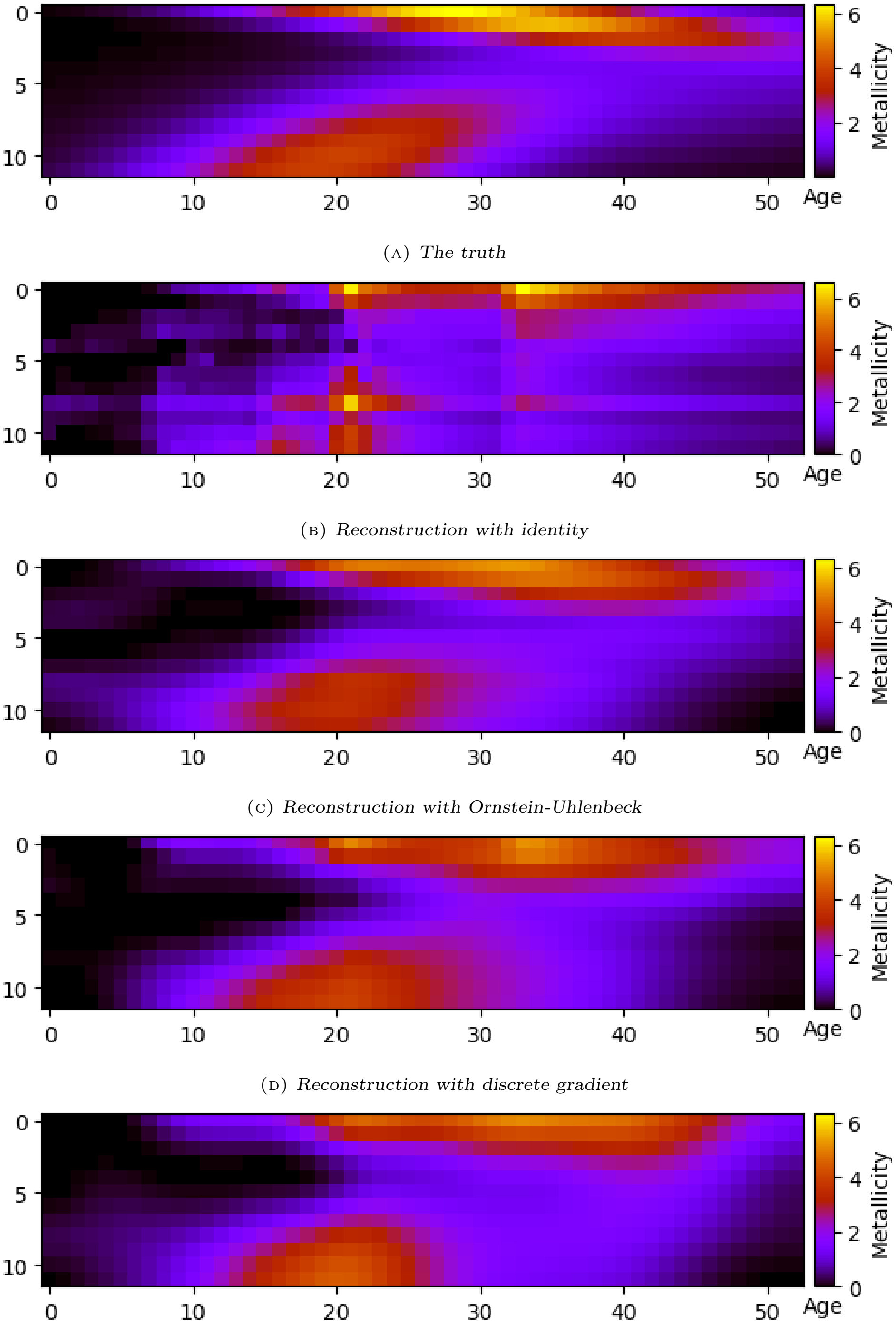
(B) *Reconstruction with identity*

(C) *Reconstruction with Ornstein-Uhlenbeck*

(D) *Reconstruction with discrete gradient*

(E) *Reconstruction with discrete Laplacian*

FIGURE 6. *The results for experiment 1 for f_2 with different regularization operators, snr = 2000.*



(A) *The truth*

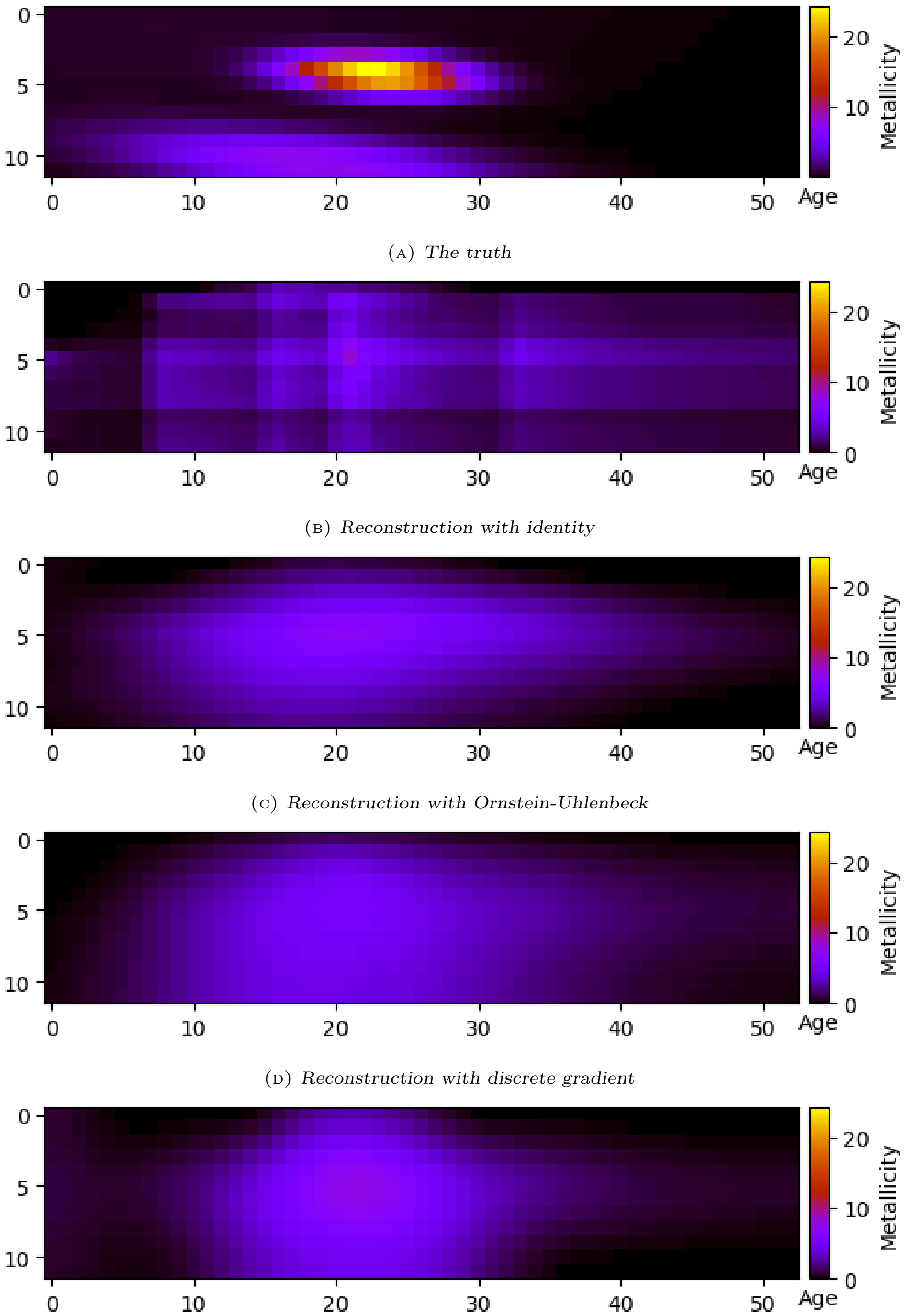
(B) *Reconstruction with identity*

(C) *Reconstruction with Ornstein-Uhlenbeck*

(D) *Reconstruction with discrete gradient*

(E) *Reconstruction with discrete Laplacian*

FIGURE 7. *The results of experiment 1 for f_3 with different regularization operators, snr =* 2000.



(A) *The truth*

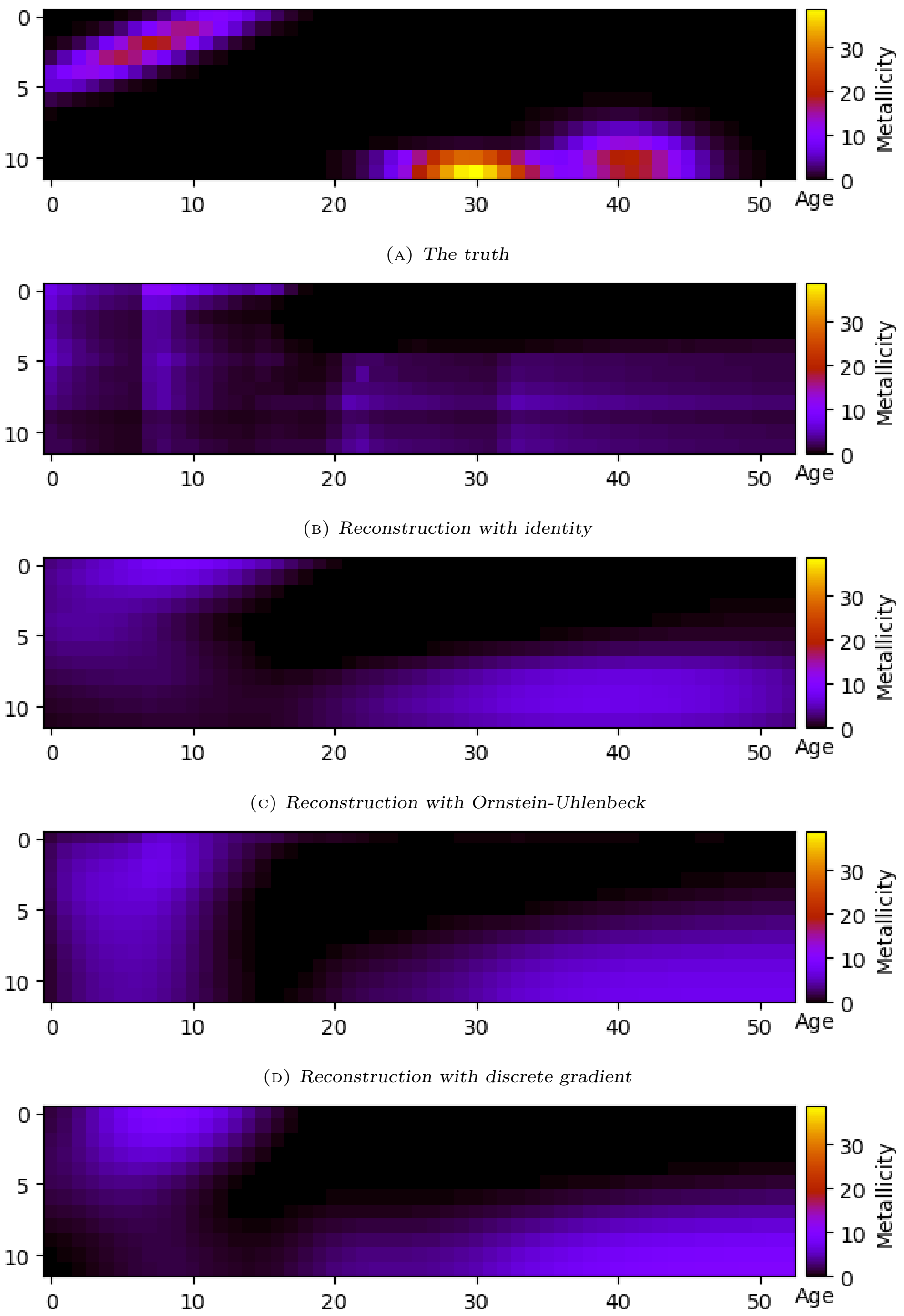
(B) *Reconstruction with identity*

(C) *Reconstruction with Ornstein-Uhlenbeck*

(D) *Reconstruction with discrete gradient*

(E) *Reconstruction with discrete Laplacian*

FIGURE 8. *The results of experiment 1 for f_1 with different regularization operators, snr = 200.*



(A) *The truth*

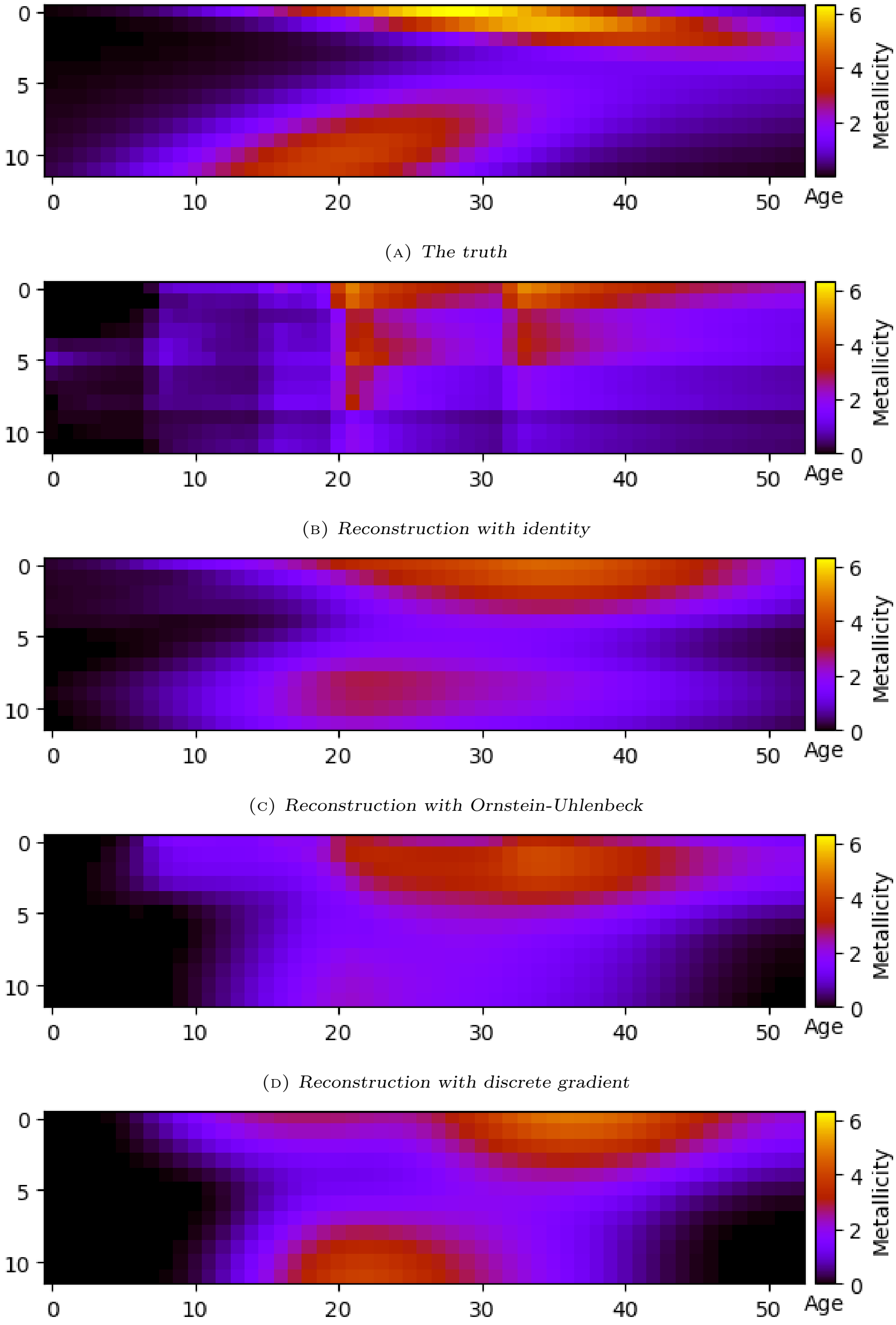
(B) *Reconstruction with identity*

(C) *Reconstruction with Ornstein-Uhlenbeck*

(D) *Reconstruction with discrete gradient*

(E) *Reconstruction with discrete Laplacian*

Figure 9. *The results of experiment 1 for f_2 with different regularization operators, snr* $= 200$.



(A) *The truth*

(B) *Reconstruction with identity*

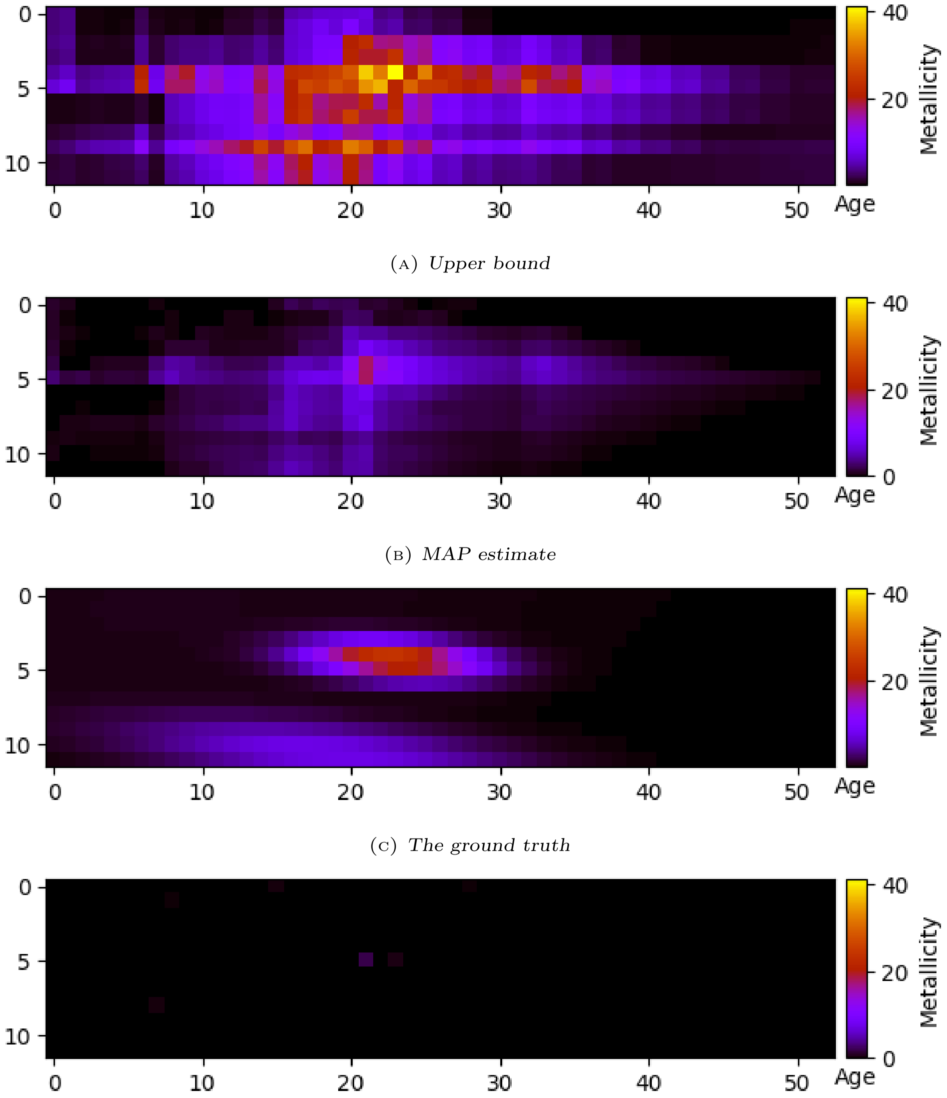
(C) *Reconstruction with Ornstein-Uhlenbeck*

(D) *Reconstruction with discrete gradient*

(E) *Reconstruction with discrete Laplacian*

FIGURE 10. *The results of experiment 1 for f_3 with different regularization operators, snr* $= 200$.

B Additional figures for experiment 4



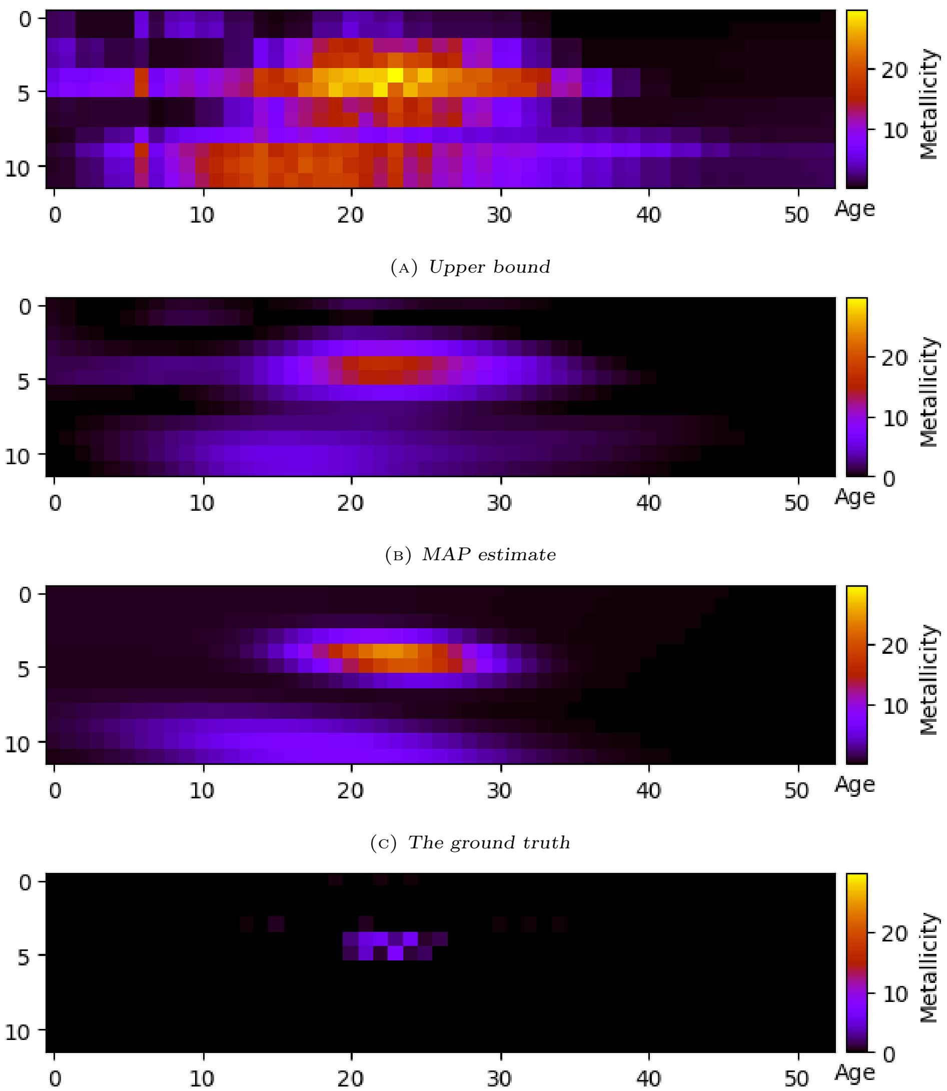
(A) *Upper bound*

(B) *MAP estimate*

(C) *The ground truth*

(D) *The lower bound*

FIGURE 11. *Uncertainty quantification with the identity covariance, snr* = 2000.



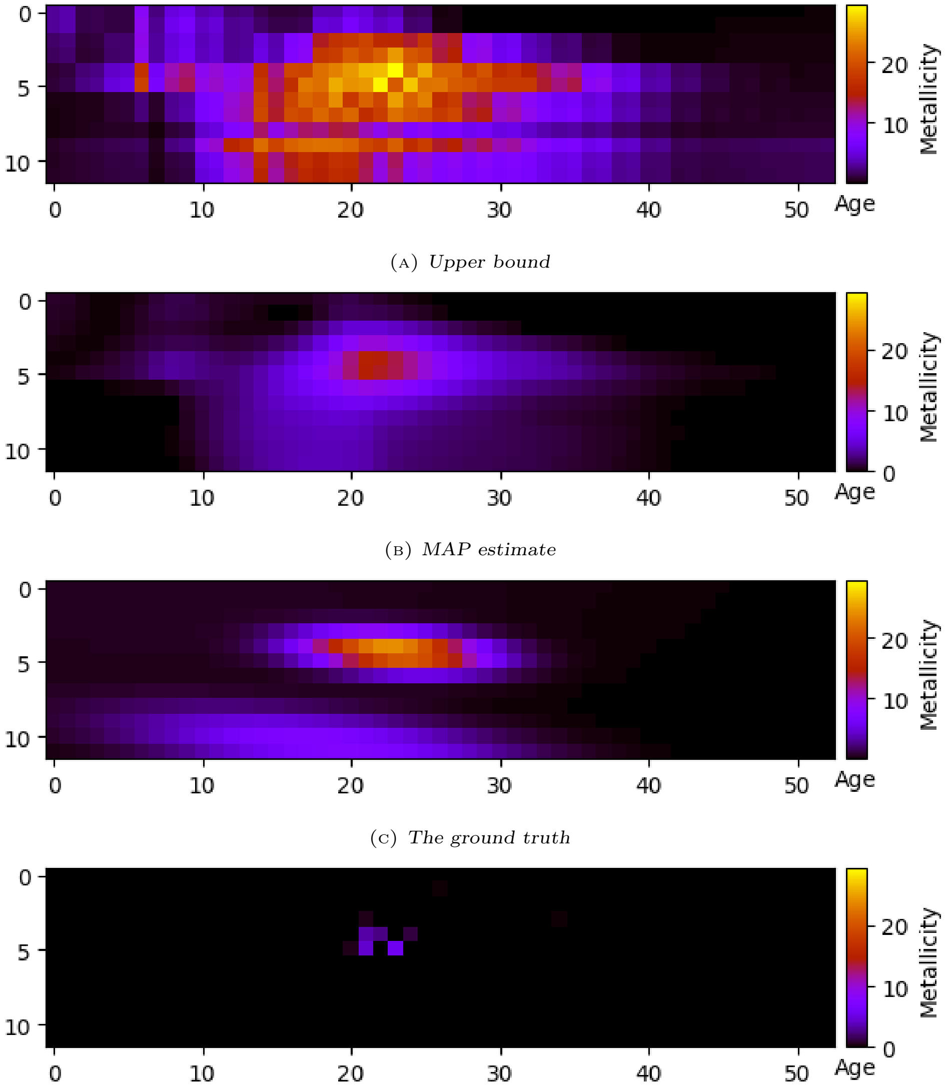
(A) *Upper bound*

(B) *MAP estimate*

(C) *The ground truth*

(D) *The lower bound*

FIGURE 12. *Uncertainty quantification with the Ornstein-Uhlenbeck covariance, snr* = 2000.



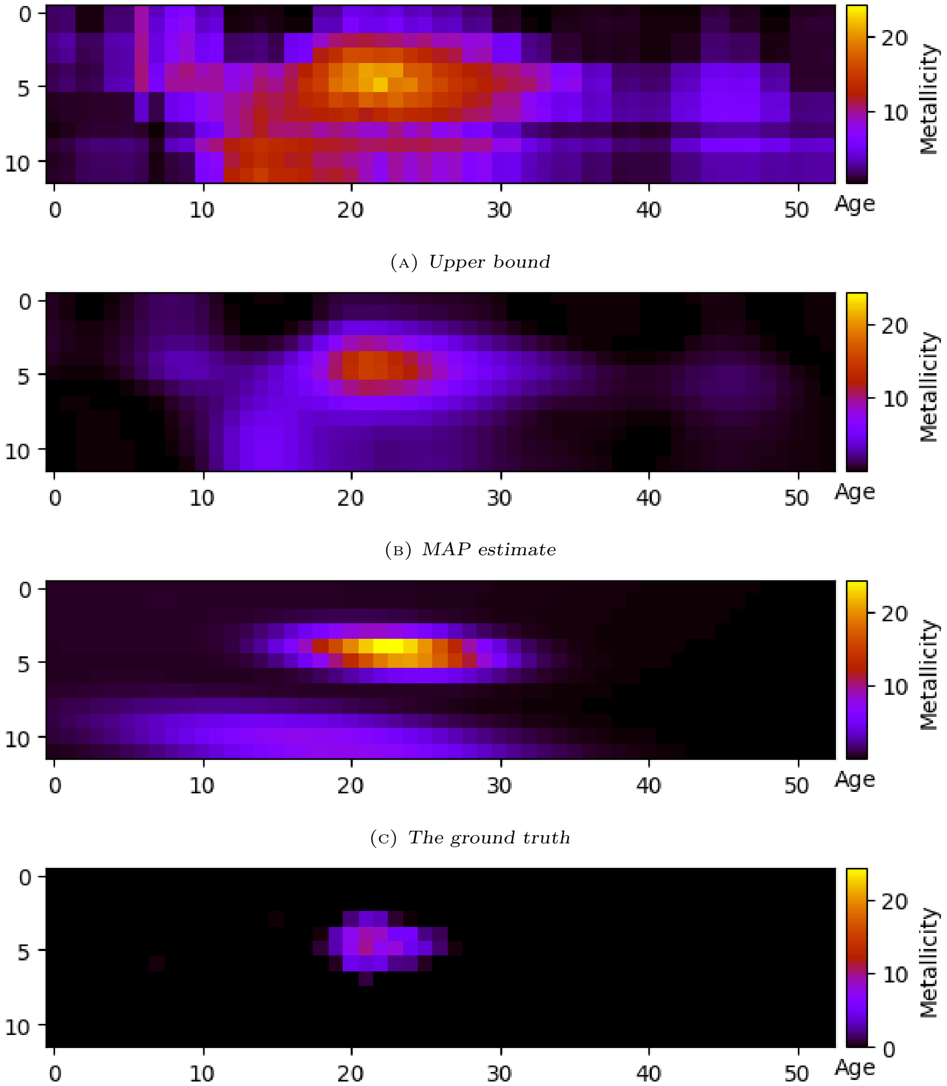
(A) *Upper bound*

(B) *MAP estimate*

(C) *The ground truth*

(D) *The lower bound*

FIGURE 13. *Uncertainty quantification with the discrete gradient regularization operator, snr* = 2000.



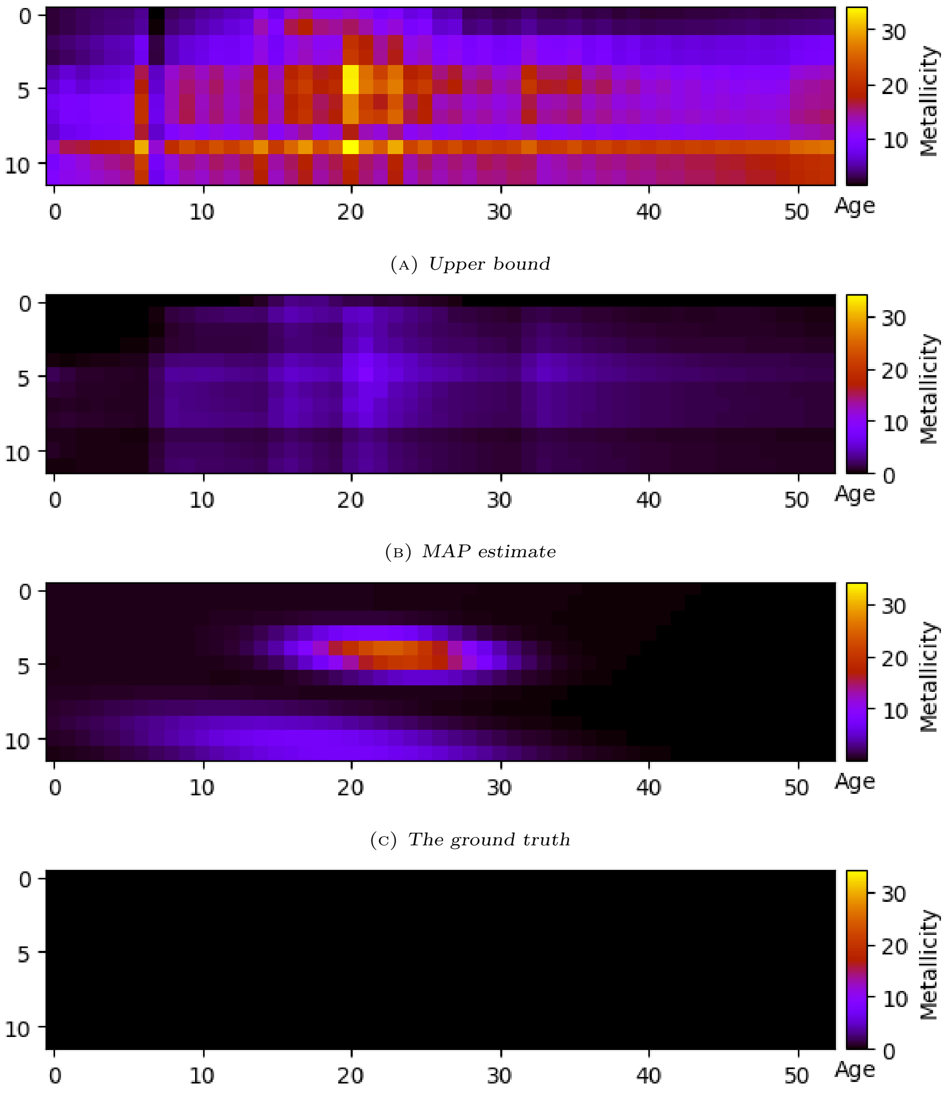
(A) *Upper bound*

(B) *MAP estimate*

(C) *The ground truth*

(D) *The lower bound*

FIGURE 14. *Uncertainty quantification with the discrete Laplacian regularization operator, snr* = 2000.



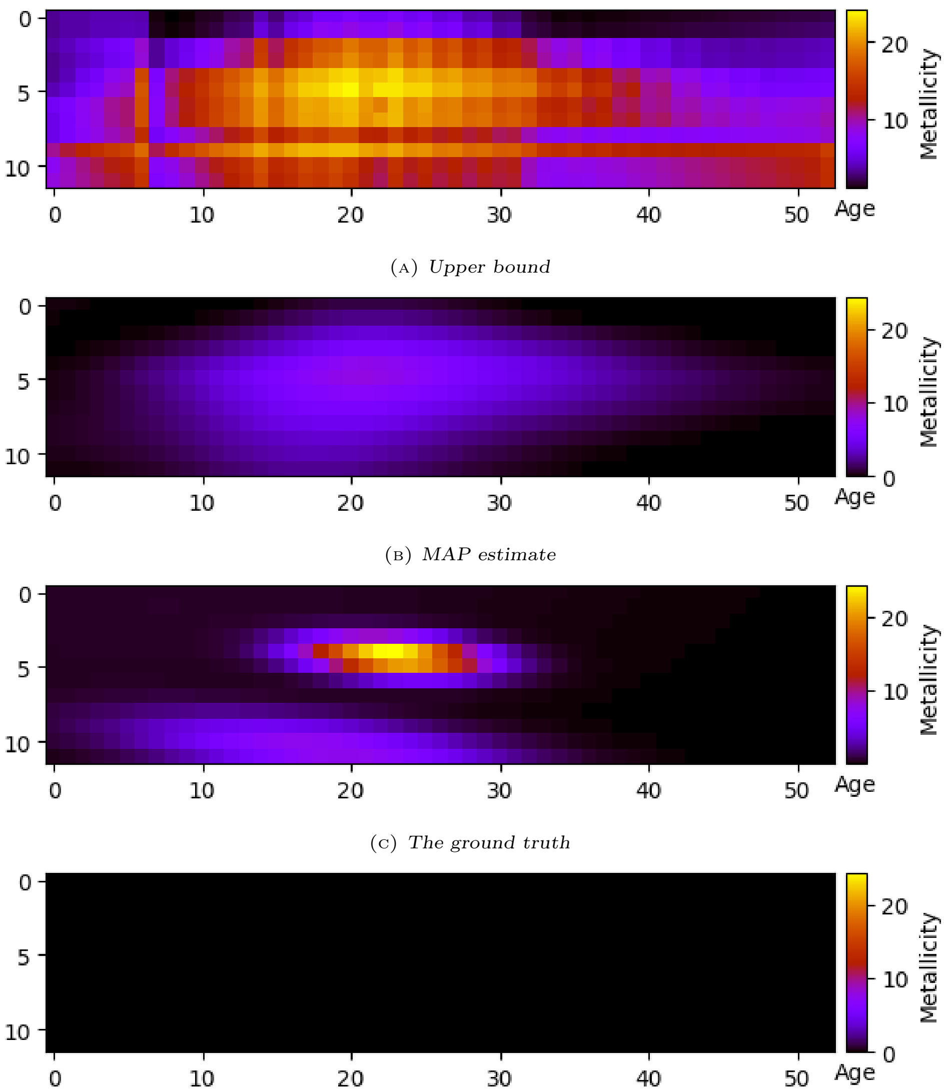
(A) *Upper bound*

(B) *MAP estimate*

(C) *The ground truth*

(D) *The lower bound*

Figure 15. *Uncertainty quantification with the identity covariance, snr* = 200.



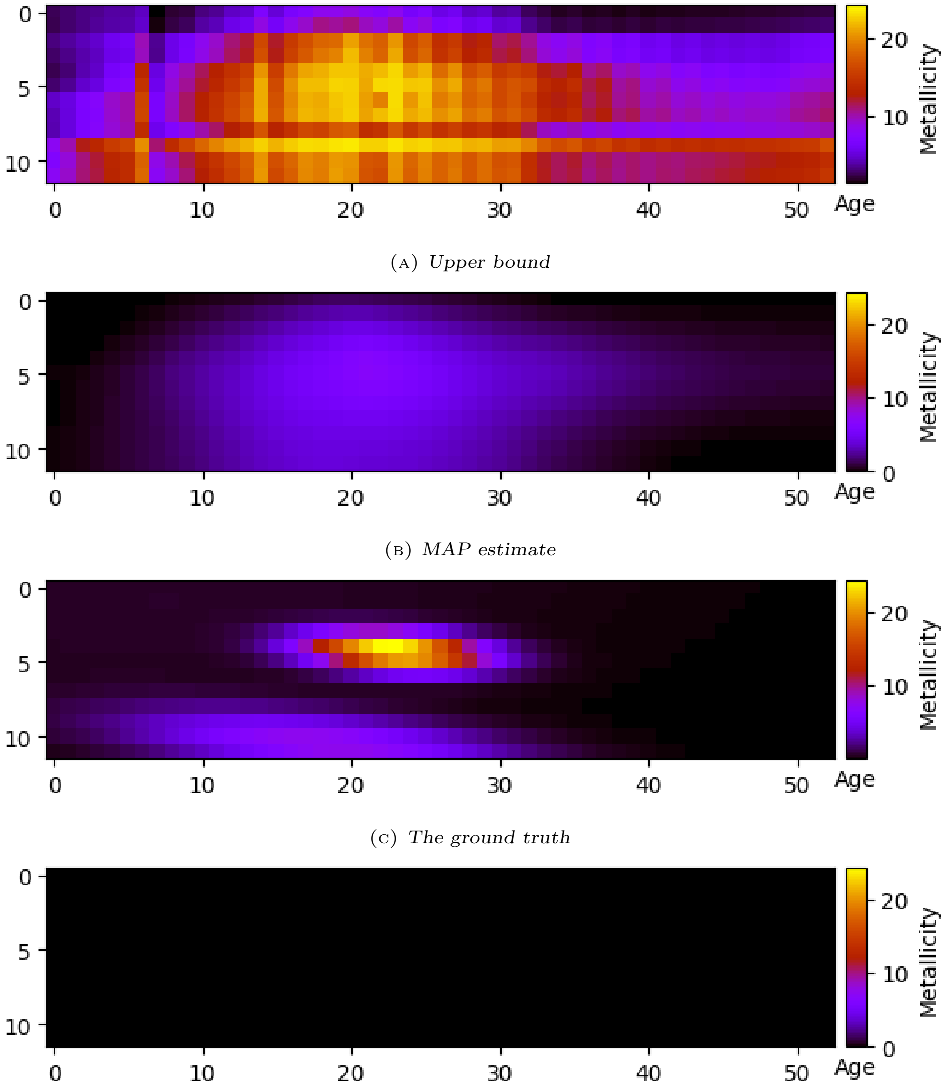
(A) *Upper bound*

(B) *MAP estimate*

(C) *The ground truth*

(D) *The lower bound*

FIGURE 16. *Uncertainty quantification with the Ornstein-Uhlenbeck covariance, snr* = 200.



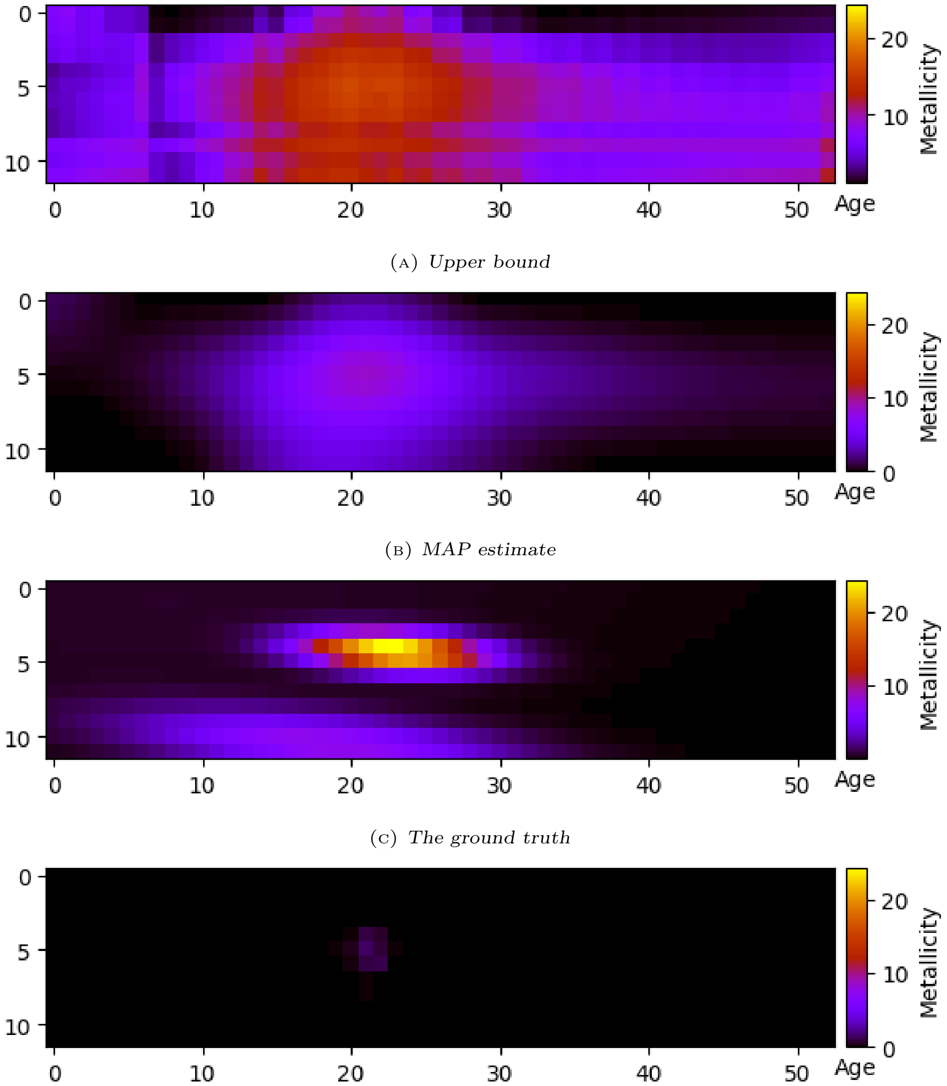
(A) *Upper bound*

(B) *MAP estimate*

(C) *The ground truth*

(D) *The lower bound*

FIGURE 17. *Uncertainty quantification with the discrete gradient regularization operator, snr* = 200.



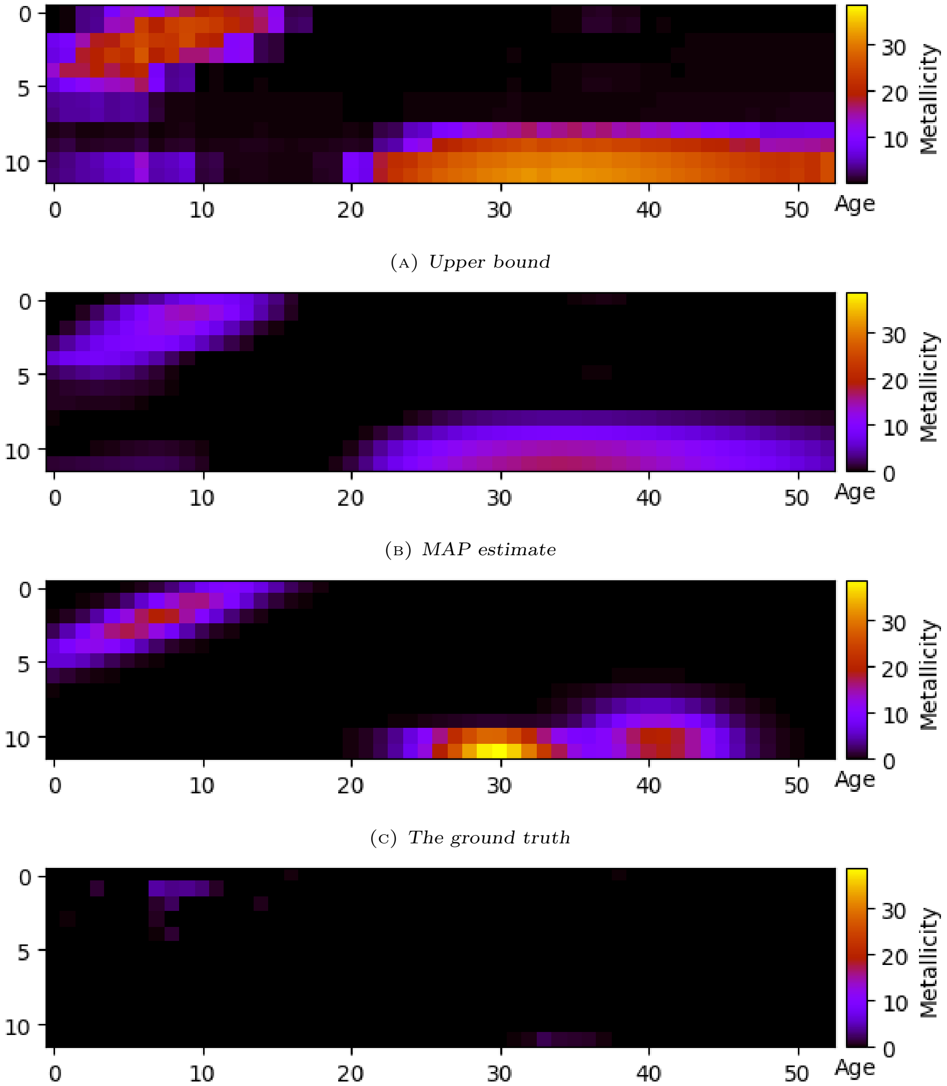
(A) *Upper bound*

(B) *MAP estimate*

(C) *The ground truth*

(D) *The lower bound*

Figure 18. *Uncertainty quantification with the discrete Laplacian regularization operator, $snr = 200$.*



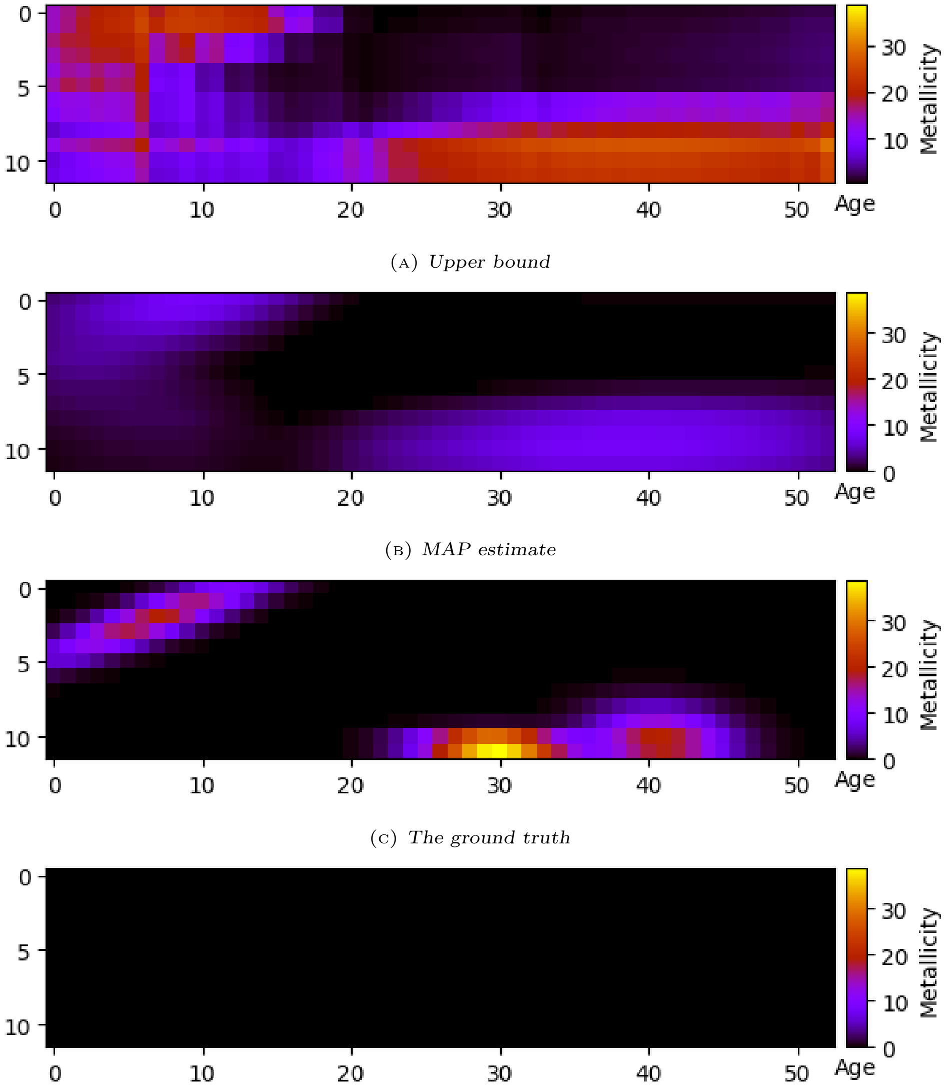
(A) *Upper bound*

(B) *MAP estimate*

(C) *The ground truth*

(D) *The lower bound*

FIGURE 19. *Uncertainty quantification with the Ornstein-Uhlenbeck covariance, test distribution f_2, snr = 2000.*



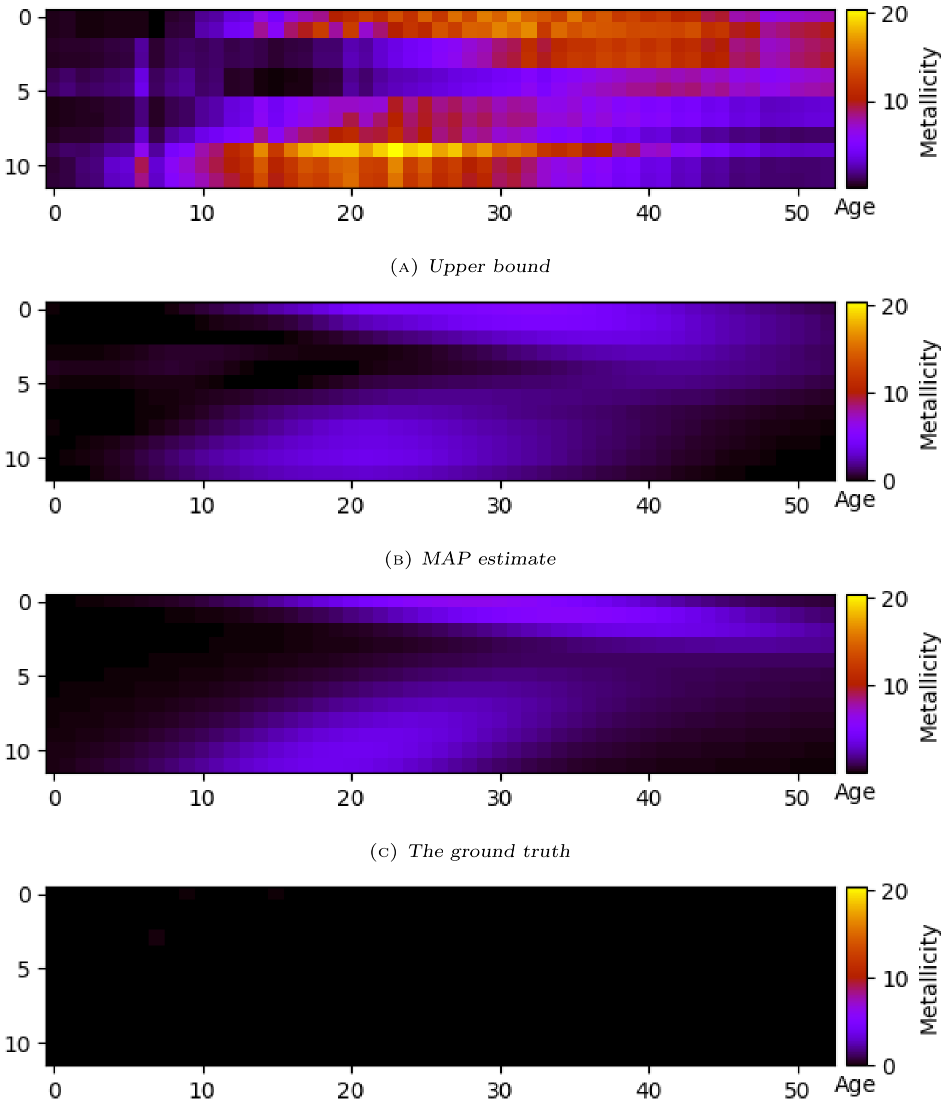
(A) *Upper bound*

(B) *MAP estimate*

(C) *The ground truth*

(D) *The lower bound*

FIGURE 20. *Uncertainty quantification with the Ornstein-Uhlenbeck covariance, test distribution f_2, snr = 200.*



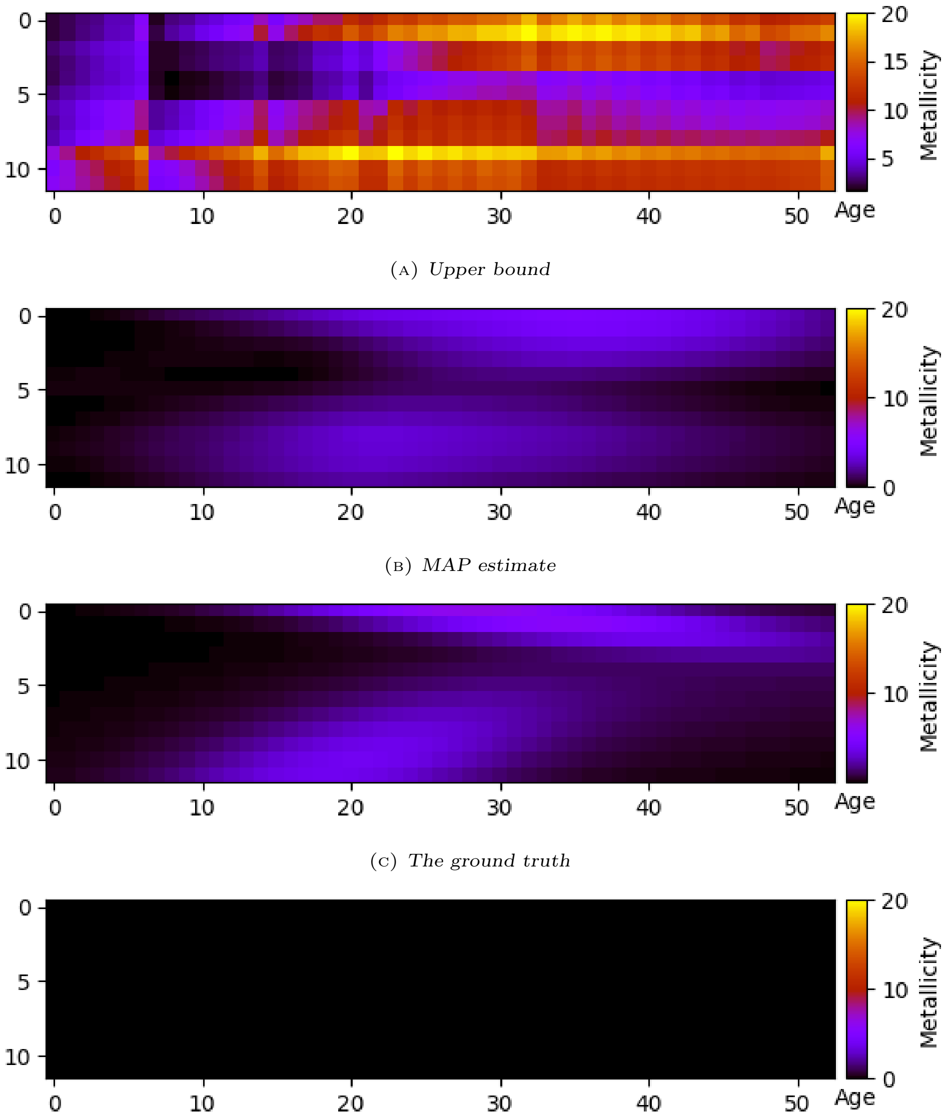
(A) *Upper bound*

(B) *MAP estimate*

(C) *The ground truth*

(D) *The lower bound*

FIGURE 21. *Uncertainty quantification with the Ornstein-Uhlenbeck covariance, test distribution f_3, snr $= 2000$.*



(A) *Upper bound*

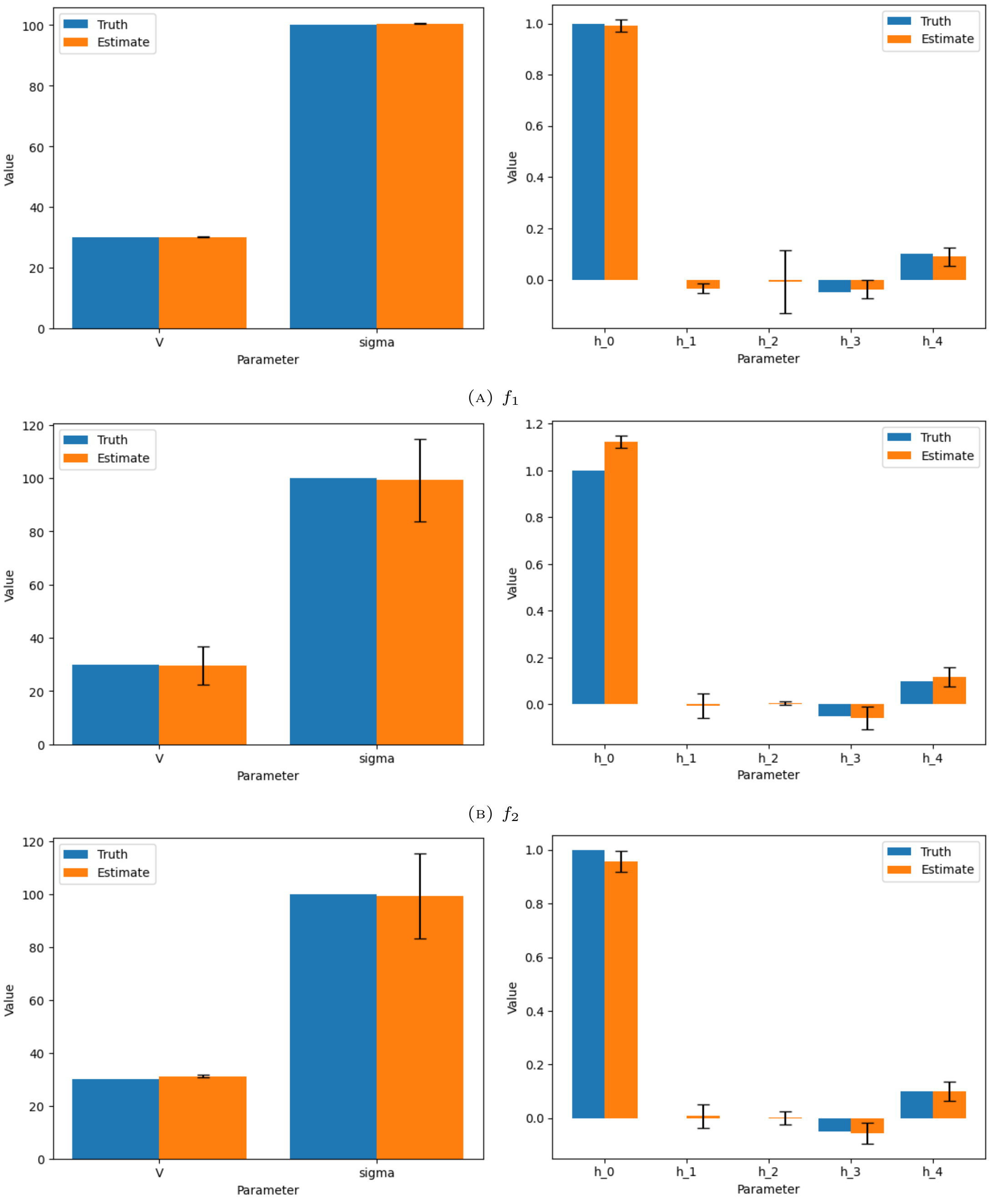
(B) *MAP estimate*

(C) *The ground truth*

(D) *The lower bound*

Figure 22. *Uncertainty quantification with the Ornstein-Uhlenbeck covariance, test distribution f_3, snr $= 200$.*

C Additional figures for experiment 5

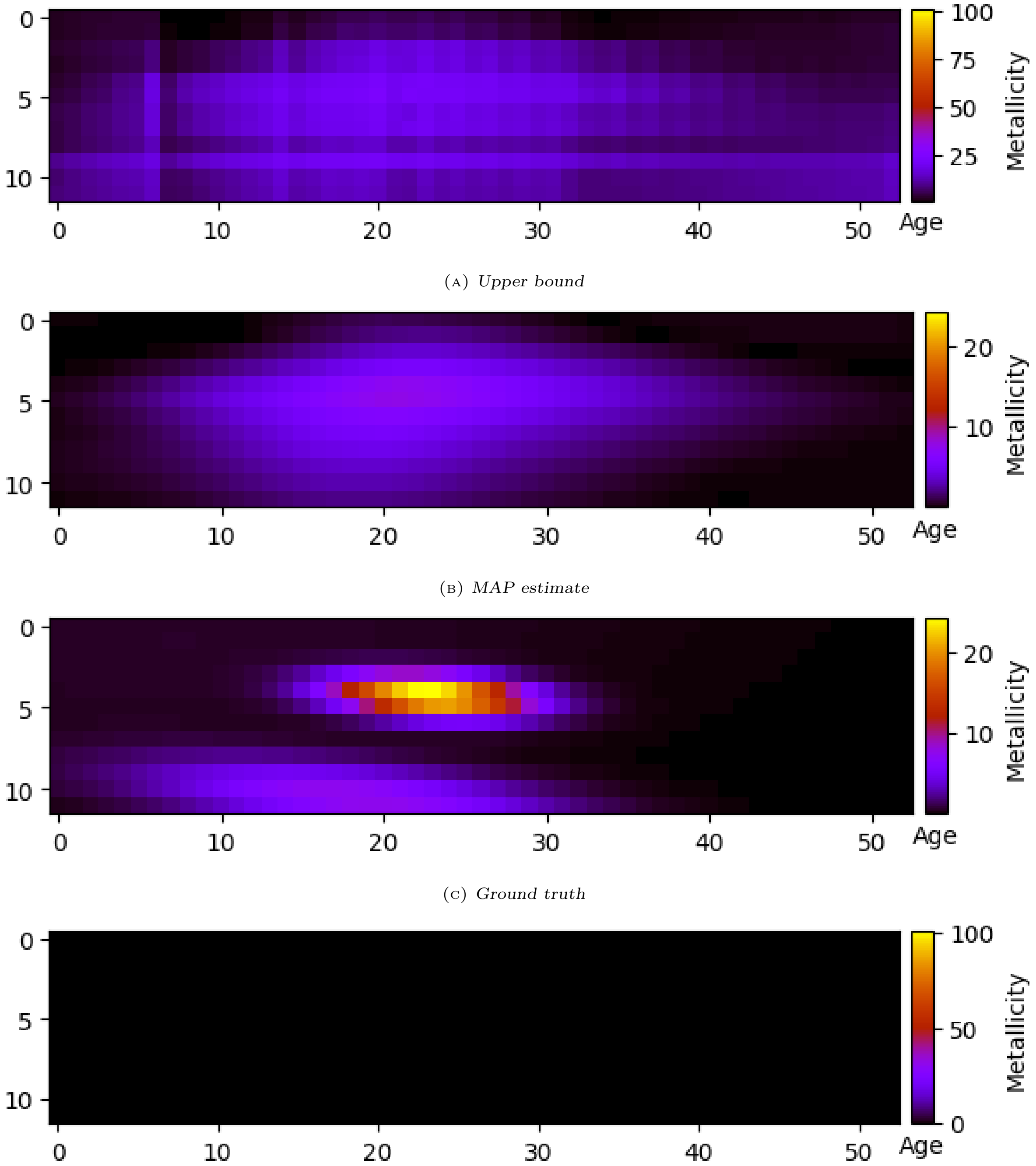


(A) f_1

(B) f_2

(C) f_3

Figure 23. *Comparison of the ground truth $\boldsymbol{\theta}_v^*$ and the MAP estimate $\hat{\boldsymbol{\theta}}_v^{\text{MAP}}$, with error bars given by the one-dimensional projections of the local credible rectangle. Note that the true value $\boldsymbol{\theta}_v^*$ is always the same, only the test functions and the initial guesses for $\boldsymbol{\theta}_v$ vary.*



(A) *Upper bound*

(B) *MAP estimate*

(C) *Ground truth*

(D) *Lower bound*

Figure 24. *Reconstruction of f_1 with $\boldsymbol{\theta}_v$ not fixed. Since the local credible rectangles are so large, the MAP and ground truth are plotted on a different scale.*

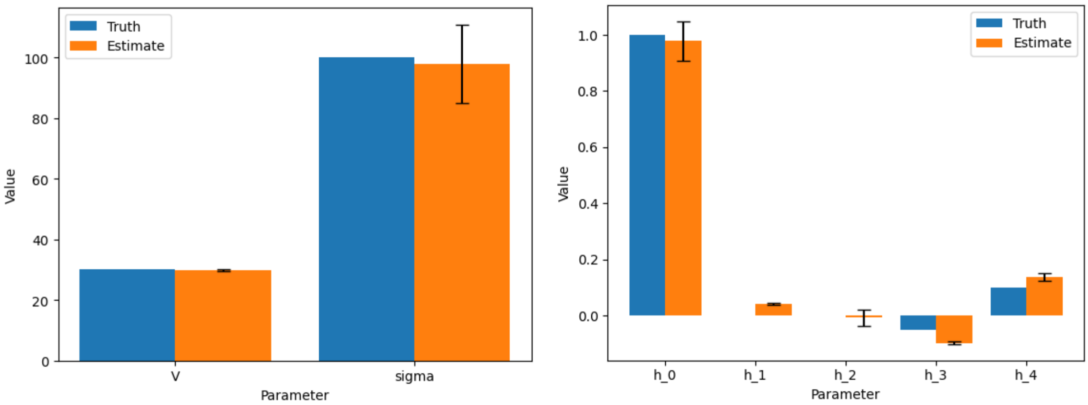


Figure 25. *Comparison of the ground truth $\boldsymbol{\theta}_v^*$ and the MAP estimate $\hat{\boldsymbol{\theta}}_v^{\mathrm{MAP}}$ for the test distribution f_1, but with regularization parameter $\beta_2 = 100$.*

References

[1] X. Cai, M. Pereyra, and J. D. McEwen. "Uncertainty quantification for radio interferometric imaging: II. MAP estimation". In: *Monthly Notices of the Royal Astronomical Society* 480.3 (2018), pp. 4170–4182. ISSN: 0035-8711. DOI: 10.1093/mnras/sty2015. URL: https://academic.oup.com/mnras/article-abstract/480/3/4170/5060771?redirectedFrom=fulltext (cited on pages 9, 10, 13).

[2] M. A. Price, X. Cai, J. D. McEwen, M. Pereyra, and T. D. Kitching. "Sparse Bayesian mass mapping with uncertainties: local credible intervals". In: *Monthly Notices of the Royal Astronomical Society* 492.1 (2019), pp. 394–404. ISSN: 0035-8711. DOI: 10.1093/mnras/stz3453. URL: https://academic.oup.com/mnras/article/492/1/394/5672642 (cited on page 10).