



# Supermodeling: The Next Level of Abstraction in the Use of Data Assimilation

Marcin Sendera<sup>1(✉)</sup>, Gregory S. Duane<sup>2,3</sup>, and Witold Dzwiniel<sup>4</sup>

<sup>1</sup> Faculty of Mathematics and Computer Science, Jagiellonian University, Kraków, Poland

[marcin.sendera@ii.uj.edu.pl](mailto:marcin.sendera@ii.uj.edu.pl)

<sup>2</sup> University of Bergen, Bergen, Norway

<sup>3</sup> University of Colorado, Boulder, CO, USA

[gregory.duane@colorado.edu](mailto:gregory.duane@colorado.edu)

<sup>4</sup> AGH-UST, Kraków, Poland

[dzwiniel@agh.edu.pl](mailto:dzwiniel@agh.edu.pl)

**Abstract.** Data assimilation (DA) is a key procedure that synchronizes a computer model with real observations. However, in the case of overparametrized complex systems modeling, the task of parameter-estimation through data assimilation can expand exponentially. It leads to unacceptable computational overhead, substantial inaccuracies in parameter matching, and wrong predictions. Here we define a *Supermodel* as a kind of ensembling scheme, which consists of a few *sub-models* representing various instances of the baseline model. The *sub-models* differ in parameter sets and are synchronized through couplings between the most sensitive dynamical variables. We demonstrate that after a short pretraining of the fully parametrized small *sub-model* ensemble, and then training a few latent parameters of the low-parameterized *Supermodel*, we can outperform in efficiency and accuracy the baseline model matched to data by a classical DA procedure.

**Keywords:** Data assimilation · Supermodeling · Dynamical systems

## 1 Introduction

Classical data assimilation (DA) procedure, which synchronizes a computer model with a real phenomenon through a set of observations, is an ill-posed inverse problem and suffers from the *curse of dimensionality* issue when used to estimate model parameters. That is, the time complexity of DA methods grows exponentially with the number of parameters and makes them helpless in the face of multiscale and sophisticated models such as models of climate&weather dynamics or tumor evolution (e.g. [12, 13, 21, 30, 37, 38]). Our idea is to assimilate data to a hierarchically organized *Supermodel*<sup>1</sup> in which the

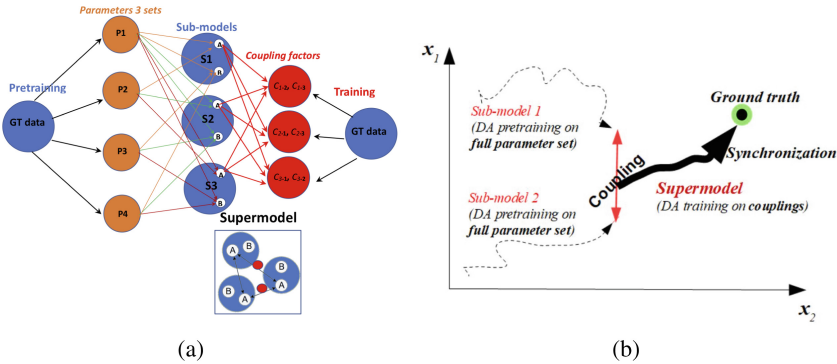
<sup>1</sup> See the *Chaos* Focus Issue introduced in [11] for the origin and history of supermodeling.

number of *trainable* metaparameters is much smaller than the number of *fixed* parameters in the *sub-models*, which themselves have to be trained in the usual schemes. We define the *Supermodel* as an ensemble of  $M$  imperfect *sub-models*  $\mu$ ,  $\mu = 1 \dots M$ , synchronized with each other through  $d$  dynamic variables and coupled to reality by observed data. Each *sub-model* is described by a set of differential equations (ordinary ones or parabolic partial ones) for the state vectors  $\mathbf{x}_\mu = (x_\mu^1, \dots, x_\mu^i, \dots, x_\mu^d)$ , such that:

$$\dot{x}_\mu^i = f_\mu^i(\mathbf{x}_\mu) + \sum_{\nu \neq \mu} C_{\mu\nu}^i (x_\nu^i - x_\mu^i) + K^i (x_{GT}^i - x_\mu^i) \quad (1)$$

$$\mathbf{x}_s(t, \mathbf{C}) \equiv \frac{1}{M} \sum_{\mu} \mathbf{x}_\mu(t, \mathbf{C}), \quad (2)$$

where the coefficients  $C_{\mu\nu}^i$  of tensor  $\mathbf{C}$  are the coupling factors synchronizing the *sub-models*,  $K$  is a set of assimilation rates “attracting” the synchronized *Supermodel* to the *ground truth* (GT) observations  $x_{GT}$ , and  $\mathbf{x}_s(\cdot)$  is the *Supermodel* output calculated as the ensemble average.



**Fig. 1.** (a) A *Supermodeling* scheme in which the *sub-models* are explicitly pre-trained and the inter-model couplings are trained without nudging the *sub-models* towards GT (as has been proposed also e.g., in [32]). We have assumed that the *sub-models* are coupled through only a single, the most sensitive, dynamical variable and the coupling factors  $C_{\mu\nu}^i$  are matched to data by using a classical DA procedure ( $K^i = 0$ ). (b) The concept of data adaptation by *Supermodeling*.

Unlike some previous applications of *Supermodeling* in climatology [29, 30], used for increasing the climate/weather forecast accuracy by relatively tight coupling ( $\mathbf{C}$  is dense) of a few very complex and heterogeneous climate models, we propose to explore *Supermodeling* from a somewhat different perspective. To this end, let us assume that the *Supermodel* is an ensemble of a few (here  $M = 3$ ) homogeneous instances of the reference (baseline) model (see Fig. 1a). The *sub-models* are represented by pretrained (e.g., using a classical DA procedure)

baseline models. This quick pretraining can be performed: (1) independently for  $M$  *sub-models*, each starting from different initial parameters or (2) exploiting  $M$  local minima of the loss function  $F(\|\mathbf{x} - \mathbf{x}_{\text{GT}}\|)$  found during initial phases of a classical DA procedure for a single, initially parametrized *sub-model*. Though the second option is more elegant and efficient computationally, we have chosen the first one to assure a greater diversity of the *sub-models*. Let us also assume that the *sub-models* are coupled through only one - the most sensitive - dynamical variable i.e.,  $\mathbf{C}$  is sparse and  $C_{\mu\nu}^i \neq 0$  only for  $i = 1$  (see Fig. 1a). In addition, we refrain from attracting the *Supermodel* to GT via the assimilation rates  $K^i$  so we assume that  $K^i = 0$  for  $i = 1 \dots d$  in Eq. 1. Instead, a classical DA algorithm (here ABC-SMC) will be employed directly for adaption of only  $\mathbf{C}$  (latent parameters) to the GT data. Because of a small number of the coupling factors  $\mathbf{C}$ , we have expected that this training procedure will be very fast. We summarize our contribution as follows:

1. We propose a novel modeling methodology, which uses the *Supermodeling* scheme as a higher level of abstraction in the use of existing DA procedures. Our approach radically speeds up the process of model training. That is, just as DA estimates states and parameters by coupling the model to a “real” system, supermodeling allows a small set of different models to assimilate data from one another; only the inter-model coupling parameters need be estimated.
2. For better synchronization of the *sub-models*, we propose their fast pretraining by employing a classical DA scheme. In previous work [13], the arbitrary parametrization of the *sub-models* often caused their desynchronization.
3. Unlike in previous *Supermodeling* proofs-of-concept, a few  $\mathbf{C}$  metaparameters can be quickly adapted to data by a classical DA method without coupling to truth. In the previous work (see, e.g. [13, 37]), non-vanishing matrices  $\mathbf{C}$  and  $K^i$  combined inter-model synchronization with a *nudging* scheme attracting the model to GT data.

In support of our modeling concept (see Fig. 1b), we present a case study: the process of parameter estimation in the Handy socio-economical model [25]. The model is a dynamical system that is an extended version of the predator-prey scheme. We have selected the Handy model due to its non-trivial behavior, reasonable computational complexity and relatively large number of parameters. On the basis of training data we try to predict the evolution of a “true” dynamical system. We compare the quality of the predictions for various time budgets for the classical ABC-SMC data assimilation method on the one hand, and the *Supermodeling* scheme on the other. Finally, we summarize and discuss the findings.

## 2 Classical Data Assimilation to the Handy Model

### 2.1 Handy Model

The Handy model is a substantial extension of the predator-prey system and is described by the time evolution of four dynamical variables: *Commoners*, *Elites*

as well as *Nature* and *Wealth* ( $x_C$ ,  $x_E$ ,  $y$ ,  $w$ ). Their evolution is described by the following equations:

$$\begin{cases} \dot{x}_C = \beta_C x_C - \alpha_C x_C \\ \dot{x}_E = \beta_E x_E - \alpha_E x_E \\ \dot{y} = \gamma y(\lambda - y) - \delta x_C y \\ \dot{w} = \delta x_C y - C_C - C_E \\ \begin{cases} C_C = \min(1, \frac{w}{w_{th}}) s x_C \\ C_E = \min(1, \frac{w}{w_{th}}) \kappa s x_E \end{cases} \\ \begin{cases} \alpha_C = \alpha_m + \max(0, 1 - \frac{C_C}{s x_C})(\alpha_M - \alpha_m) \\ \alpha_E = \alpha_m + \max(0, 1 - \frac{C_E}{s x_E})(\alpha_M - \alpha_m) \end{cases} \\ w_{th} = \rho x_C + \kappa \rho x_E \end{cases} \quad (3)$$

In Table 1 we have compiled a glossary of parameters and variables, and their *ground truth* or initial values, respectively. In Fig. 2 we illustrate the typical evolution of the dynamical variables of the Handy model. The time evolution of the system is so variable and its parameters so sensitive that prediction of the model behavior is sufficiently difficult as to make data assimilation a non-trivial task.

**Table 1.** Parameters and initial values of dynamical variables of the Handy model.

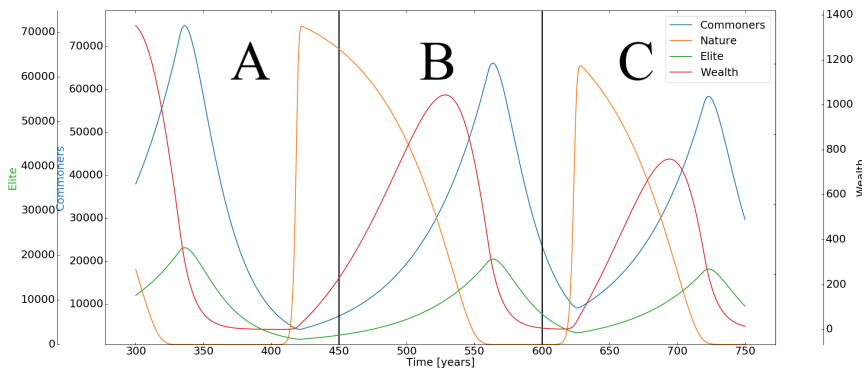
Parameter	Description	Value
$\alpha_m$	Normal (minimum) death rate	$1.0 \times 10^{-2}$
$\alpha_M$	Famine (maximum) death rate	$7.0 \times 10^{-2}$
$\beta_C$	Commoners birth rate	$3.0 \times 10^{-2}$
$\beta_E$	Elites birth rate	$3.0 \times 10^{-2}$
$s$	Subsistence salary per capita	$5.0 \times 10^{-4}$
$\rho$	Threshold wealth per capita	$5.0 \times 10^{-3}$
$\gamma$	Regeneration rate of nature	$1.0 \times 10^{-2}$
$\lambda$	Nature carrying capacity	$1.0 \times 10^2$
$\kappa$	Inequality factor	1.0
$\delta$	Depletion (production) factor	3.34
Variable	Description	Initial value
$x_C$	Commoners population	$1.0 \times 10^2$
$x_E$	Elites population	$2.9 \times 10^1$
$y$	Nature	$1.0 \times 10^2$
$w$	Accumulated wealth	$5.0 \times 10^1$

## 2.2 Ground Truth Data Generation

To further the testing of the supermodel concept, we generated artificial data, assuming that there exists a *ground-truth* “model” that simulates reality. Of course, because neither reality nor observations of reality can be accurately approximated by any mathematical model, we should somehow disturb both observations and the whole model as well. The comparison of the robustness of ABC and *Supermodeling* by using such a stochastic model would need many extensive tests. Nevertheless, conducting such research would make sense if the *Supermodeling* scheme outperforms a classical DA procedure for a much simpler *ground truth* model. Therefore, herein we have assumed that reality follows exactly a given baseline mathematical model with a rigid and “unknown” set of parameters. Our role is to guess them, having a limited number of observations, i.e., samples from this GT system evolution.

As presented in Fig. 2, the dynamical variables of the GT model evolve in a given time interval in a smooth but variable and non-trivial way. We consider here only one time interval (from  $T_1 = 300$  up to  $T_2 = 750$  timesteps) that was split into three subintervals of the same length ( $A = [300, 450]$ ,  $B = [450, 600]$ ,  $C = [600, 750]$ ). The models (the baseline model, *sub-models* and *Supermodel*) will be trained on GT data sampled in the middle part  $B$  of the plot, and accuracies of predictions will be tested on  $A$  (backward forecasting),  $C$  (forward forecasting) and  $A \cup C$  (overall) time intervals. We have decided to use both sparsely and densely sampled data, i.e., in each of the training subintervals we have generated “real” observations every  $\Delta T_1 = 10$  or  $\Delta T_2 = 3$  steps, respectively.

In the rest of this paper we present the results from the case study of data assimilation to the Handy model and arbitrarily selected fragments of its behavior (Fig. 2). We have tested our approach on other datasets, from which the same conclusions can be drawn. Some results and all numerical details can be found in the MSc thesis [31].



**Fig. 2.** The behaviour of the Handy model as the *ground-truth* model in selected time intervals. A fragment of the model time evolution is divided into three intervals (A, B, C), wherein the middle one B is used for generation of the training data while the remaining two for the test data.

### 2.3 Sensitivity Analysis

In many data assimilation tasks, knowledge of the most sensitive model parameters and dynamic variables, can help to give a faster and more precise search of the parameter space. This is particularly true if expert knowledge is unavailable. In the context of *Supermodeling*, the most sensitive dynamic variable has to be identified for use in synchronizing the sub-models. To determine the most significant dynamical variable, we performed Sobol Sensitivity Analysis (SA) [27,34]. Herein, we use the society quality measure:

$$Q = \frac{w}{x_C + x_E}, \quad (4)$$

to calculate the Sobol indices, where  $x_C$ ,  $x_E$  are the populations of *Commoners* and *Elites* respectively, and  $w$  is the society's overall *Wealth*. We estimated that *Elites* is the most sensitive dynamical variable, also because it is closely connected with the most sensitive parameter,  $\beta_E$ , the Elites' birth rate (see Table 2). However, the SA procedure might be skipped if the most sensitive variable is already known, e.g., due to *a priori* possession of expert knowledge.

**Table 2.** The Sobol sensitivity indices  $S_1$  and  $S_T$  for the parameters and dynamical variables of the Handy model (a greater value of the index means higher sensitivity).

Parameter	Description	$S_1$	$S_T$
$\alpha_m$	Normal (minimum) death rate	$2.4 \times 10^{-2}$	$3.0 \times 10^{-1}$
$\alpha_M$	Famine (maximum) death rate	$1.6 \times 10^{-3}$	$4.6 \times 10^{-2}$
$\beta_C$	Commoners birth rate	$8.3 \times 10^{-2}$	$4.6 \times 10^{-1}$
$\beta_E$	Elites birth rate	$1.1 \times 10^{-1}$	$4.8 \times 10^{-1}$
$s$	Subsistence salary per capita	$3.3 \times 10^{-3}$	$3.1 \times 10^{-2}$
$\rho$	Threshold wealth per capita	$5.8 \times 10^{-4}$	$2.1 \times 10^{-3}$
$\gamma$	Regeneration rate of nature	$2.5 \times 10^{-3}$	$3.2 \times 10^{-2}$
$\lambda$	Nature carrying capacity	$5.8 \times 10^{-3}$	$1.0 \times 10^{-1}$
$\kappa$	Inequality factor	$5.9 \times 10^{-2}$	$4.2 \times 10^{-1}$
$\delta$	Depletion (production) factor	$2.4 \times 10^{-2}$	$2.7 \times 10^{-1}$
Variable (initial value)	Description	$S_1$	$S_T$
$x_C(0)$	Commoners population	$1.9 \times 10^{-3}$	$1.0 \times 10^{-2}$
$x_E(0)$	Elites population	$4.5 \times 10^{-3}$	$2.1 \times 10^{-2}$
$y(0)$	Nature	$4.5 \times 10^{-7}$	$5.9 \times 10^{-9}$
$w(0)$	Accumulated wealth	$2.6 \times 10^{-3}$	$3.2 \times 10^{-4}$

### 2.4 Approximate Bayesian Computation

Approximate Bayesian Computation (ABC) is not a single algorithm, but rather a very wide class of algorithms and methods that employ Bayesian inference

for data assimilation purposes [7, 14]. The main novelty of these methods is in their correct estimation of parameters even when the likelihoods are intractable [36]. In ABC algorithms, the functions of likelihood are not calculated, but the likelihood is approximated by the comparison of observed and simulated data [36].

Let us assume, that  $\theta \in \mathbb{R}^n, n \geq 1$  is a vector of  $n$  parameters and  $p(\theta)$  is a prior distribution. Then the goal of ABC approach is to approximate the posterior distribution  $p(\theta|D)$  where  $D$  is the real data [1]. The posterior distribution is approximated in the following way:

$$p(\theta|D) \propto f(D|\theta)p(\theta), \quad (5)$$

where  $f(D|\theta)$  is the function of likelihood of  $\theta$  given the dataset  $D$  [35].

Among the variety of different approaches, one of the most useful is the ABC-SMC algorithm that uses the sequential Monte Carlo (M-C) method [7]. The major novelty, in comparison with previous methodologies (e.g., ABC-MCMC [24]), is the introduction of a set of *particles*  $\theta^{(1)}, \dots, \theta^{(S)}$  (parameter values sampled from a prior distribution  $p(\theta)$ ), used to produce a sequence of intermediate distributions  $p(\theta|d(D, \tilde{D}) \leq \epsilon_i)$  (for  $i = 1, \dots, T - 1$ ) [36]. The *particles'* M-C propagation stops when a good representation of the target distribution ( $p(\theta|d(D, \tilde{D}) \leq \epsilon_T)$ ) is achieved. The set of error tolerance thresholds is chosen to be a decreasing sequence  $\epsilon_1 > \dots > \epsilon_T \geq 0$  that ensures the convergence of the intermediate probability distributions (of the parameters values) to the target ones. In the ABC-SMC algorithm, the parameter perturbation kernel can be simulated by the random walk procedure, with Gaussian or uniform functions [36]. Simultaneously, an adequately large set of *particles* will allow the Markov process to avoid low-probability regions and local minima in the parameter space.

## 2.5 ABC-SMC Training Results

For training the Handy model we use the ABC-SMC algorithm assuming that:

1. the number of *particles*,  $S = 100$ ;
2. we fix the training time,  $t_{max}$ ;
3. we set intervals of possible values of parameters to be  $\pm 10\%$  of exact (*ground truth*) ones (see Table 1);
4. the cost function is the root-mean-square error (RMSE).

Thus we assume that we know some approximate values of parameters. However, in the future we should also investigate the robustness of ABC-SMC and *Supermodeling* against prior selection of the values of the *sub-models'* parameters. In Table 3A, and Table 3B we present the training time (CPU time) and the RMSE errors of predictions for the Handy model for two pre-defined training error goals: RMSE = 50 and 100, respectively. The timings were measured for the layout presented in Fig. 2. We observe more than a ten-fold increase of the computational time when RMSE training precision goes from 100 to 50 (in dimensionless units) for sparsely sampled data. But for denser sampling, this increase is only two-fold.

We do not observe, in either case, any increase of the overall prediction quality with training precision. Meanwhile, one can notice signs of overfitting. A small increase is observed only for *forward prediction* (C). However, this improvement does not compensate the substantial decrease in *backward prediction* (A) quality. Summing up, both decreasing the training error and increasing the sampling frequency may lead to overfitting, so careful design is needed.

**Table 3.** The averaged CPU times and the prediction accuracies for the ABC-SMC training, to achieve given errors (RMSE = 50 and 100) with the *ground truth* training data. Results for sparser  $\Delta T_1 = 10$  and denser  $\Delta T_2 = 3$  sampled data.

Sampling	$\Delta T_1 = 10$				$\Delta T_2 = 3$			
Training error [RMSE]	Training time [s]	Overall prediction [RMSE]	Backward prediction [RMSE]	Forward prediction [RMSE]	Training time [s]	Overall prediction [RMSE]	Backward prediction [RMSE]	Forward prediction [RMSE]
50.0	1894.89	1387.58	1337.08	1369.54	248.51	1402.46	1303.10	1436.00
100.0	162.29	1303.83	1012.57	1456.40	111.74	1335.02	915.79	1559.29

### 3 Supermodeling the Handy System

#### 3.1 Supermodeling by Data Assimilation Between Models

The *Supermodeling* approach is described in detail in the Introduction. Below we enumerate the main steps.

1. Create a small number  $M$  of instances (the *sub-models*) of the baseline model, initializing their parameters with a *rule-of-thumb* and/or using expert knowledge.
2. Pretrain every *sub-model*  $\mu = 1, \dots, M$  by using a classical DA procedure on the samples from Fig. 2B. New parameter sets will thus be generated for each *sub-model*.
3. Create the *Supermodel* by coupling the ODEs from Eqs. 3 through the most sensitive dynamical variable, as in Eq. 1, but with  $K^i = 0$  and  $C_{\mu\nu}^i = 0$  for  $i \neq 1$  (Fig. 1a).
4. Train the coupling factors  $C_{\mu\nu}^i$  of the *Supermodel* on the sampled data from Fig. 2B, according to the scheme sketched below, until either the RMS error relative to GT falls below a designated value or the elapsed training time reaches  $t_{max}$ .
5. The *Supermodel* trajectory is defined by averaging the sub-models states (Eq. 2).

#### 3.2 Training Details

Unlike the classical DA training scheme described in Sect. 2.5, we fix not only the maximum time  $t_{max}$  but also the time needed for pretraining each of the *sub-models*  $t_{sub}$ . We pretrain  $M = 3$  *sub-models* with ABC-SMC (one by one, each



for a time  $t_{sub}$ ) and couple them via the most sensitive variable  $x_E$ , to form the *Supermodel*. We also restrict the coupling coefficients to a fixed interval  $[0, 0.5]$  (as in [12]). Furthermore, to speed-up the DA process, we divide the training data from Fig. 2B into five subintervals (*mini batches*) of the same length. Finally, we train the *Supermodel* with the ABC-SMC algorithm on the sequence of *mini batches* one after another for the estimated time  $t_{sumo} = t_{max} - \overline{t_{sub}}$  (where  $\overline{t_{sub}}$  is the mean time of pretraining the *sub-models*). Because the processes of pretraining the *sub-models* are independent, we have assumed that they are calculated in parallel. Then the normalized time for the *Supermodel* training will be equal to  $t_{max}$ .

We have performed the computations on the Prometheus supercomputer located in the ACK Cyfronet AGH UST, Krakow, Poland. We have used just one node, that consists of 8 CPUs (Intel Xeon E5-2680 v3, 2.5 GHz) with 12 cores each, giving 96 computational cores in total.

### 3.3 Results

Here we compare the *Supermodeling* scheme with the ABC-SMC DA algorithm with four different time budgets  $t_{max}$ : 14, 50, 100 and 250 s. Toward this end, we have constructed several *Supermodels*, each consisting of  $M=3$  differently initialized *sub-models*. Each *sub-model* was pretrained for a given short time period  $t_{sub} < t_{max}$ . We have selected several combinations, constructing four *Supermodels* which differ in the *sub-models*' pretraining time. We have repeated *Supermodel* training and testing procedure ten times for each pair  $(t_{max}, t_{sub})$  and for various parameter initializations. Next, we have removed *zeroth* and *tenth* 10-quantiles from the results. The RMSE values on the test set (*backward prediction*, *forward prediction* and *overall prediction*) were averaged and the standard deviation was calculated. We present these averages for both sparsely ( $\Delta T_1 = 10$ ) (see Table 4A) and densely ( $\Delta T_2 = 3$ ) sampled datasets (see Table 4B).

As shown in Table 4A and Table 4B, the *forward prediction* RMSE error is a few times smaller for two-stage *Supermodeling* than for the classical parameter estimation with the ABC-SMC algorithm, for both sparse and denser datasets, and for all time regimes. Furthermore, with the ABC-SMC algorithm, longer learning appears to cause overfitting. It is important to mention that the ABC-SMC algorithm reaches the minimum RMSE after about 70 s of training. (The minimum is flat up to 120 s and afterwards RMSE grows due to overfitting.) Therefore, for *Supermodel*.70, composed of *sub-models* pretrained in 70 s, we obtain a radically lower RMSE, as compared to that for ABC-SMC, as total training time increases.

Turning attention to *backward prediction*, we note that although the *Supermodeling* approach is still convincingly better for overall prediction (except in one case) than the classical DA algorithm, the advantage for backward prediction is not so radical as for *forward prediction*. This bias can be clearly seen in Fig. 3 and Fig. 4, particularly, for the normalized RMSE plot. This behaviour is not seen with the ABC-SMC algorithm. It is the result of the specific training procedure we employed for the *Supermodeling* algorithm. The algorithm is

**Table 4.** RMS errors for the ABC algorithm and for the supermodel, for sparser (A)  $\Delta T_1 = 10$ , and denser (B)  $\Delta T_2 = 3$  datasets. Supermodel\_ $\{X\}$  is the *Supermodel* with the *sub-model* pretraining time  $t_{sub}$  set to  $X$  seconds. The better result for each case is shown in bold.

Sampling	$\Delta T_1 = 10$				$\Delta T_2 = 3$			
Time regime [s]	Method	Overall prediction [RMSE]	Backward prediction [RMSE]	Forward prediction [RMSE]	Method	Overall prediction [RMSE]	Backward prediction [RMSE]	Forward prediction [RMSE]
14 s	ABC-SMC	4586.38	4273.77	4879.00	ABC-SMC	5708.75	5181.07	6191.62
	Supermodel <sub>10</sub>	<b>1897.73</b>	<b>2497.55</b>	<b>982.33</b>	Supermodel <sub>12</sub>	<b>2453.95</b>	<b>3128.33</b>	<b>1502.42</b>
50 s	ABC-SMC	<b>1238.31</b>	<b>1213.21</b>	1262.92	ABC-SMC	2543.26	<b>2436.65</b>	2645.58
	Supermodel <sub>35</sub>	1710.62	2350.98	<b>570.37</b>	Supermodel <sub>20</sub>	<b>1959.12</b>	2652.73	<b>799.61</b>
100 s	ABC-SMC	1153.41	947.22	1327.96	ABC-SMC	1315.0	<b>930.0</b>	1510.0
	Supermodel <sub>70</sub>	<b>665.46</b>	<b>762.62</b>	<b>551.45</b>	Supermodel <sub>60</sub>	<b>855.24</b>	1098.18	<b>506.83</b>
250 s	ABC-SMC	1380.90	1511.72	1236.32	ABC-SMC	1410.4	1355.1	1401.0
	Supermodel <sub>70</sub>	<b>466.33</b>	<b>548.09</b>	<b>366.78</b>	Supermodel <sub>70</sub>	<b>600.08</b>	<b>795.76</b>	<b>294.87</b>

trained in five mini-batches starting from the left-hand side of the training interval (Fig. 2B). Consequently, the fitting accuracy is highest at the right-hand side of the B interval. At the last training point ( $t = 600$ ) the standard deviation is equal to 0, while at the first point ( $t = 450$ ) it is distinctly greater.

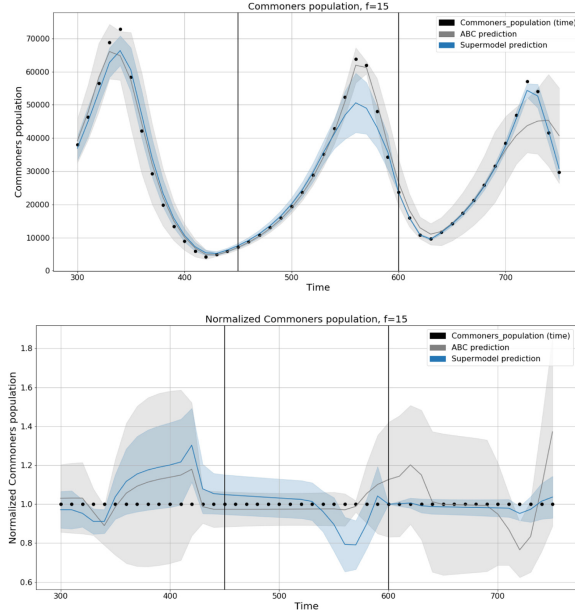
In summary, we conclude that the *Supermodeling* scheme results in predictions closer to the actual time series and with lower uncertainties, especially, for the *forward prediction* task. We have observed similar effects for other data, as presented in [31].

## 4 Discussion and Related Work

Classical data assimilation procedures were formulated on the basis of variational and Bayesian frameworks [2, 26]. The existing DA algorithms can be divided onto two main groups: (1) sequential-Monte-Carlo-based (e.g., [3]) and (2) Kalman-filter-based methods (e.g., [28])<sup>2</sup>, which have formed the core of many other DA algorithms (e.g., [5]). Over the years, the majority of research in this direction was focused primarily on the improvement of the predictions’ accuracy on tasks ranging from small-scale problems (e.g. [20]) to weather prediction [18]. Recently, more and more studies have attempted to speed-up data assimilation methods and to enable their use with extremely complex multi-scale models (e.g., [19, 26]).

The greatest challenge that arises with sequential Monte-Carlo-based methods (i.e., the ABC-SMC algorithm), is the requirement that a very large number of simulations need be performed, especially for the inverse problem of estimating parameters. That is, parameters can be adjoined to the model state and treated as variable quantities to be estimated - the second level of abstraction in the use of DA. But the number of required simulations increases exponentially with the number of model parameters (see e.g. [17]). To outperform the classical DA

<sup>2</sup> Kalman filtering is equivalent to the popular 4D-Var algorithm, for a perfect model.

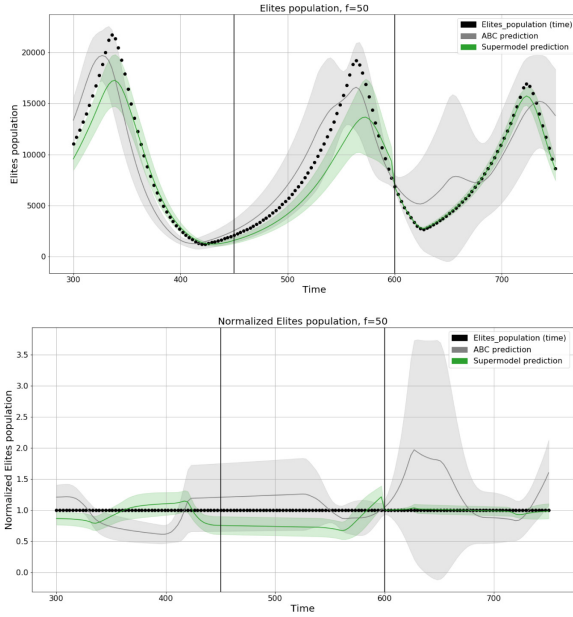


**Fig. 3.** The results for sparsely ( $\Delta T_1 = 10$ ) sampled data and  $t_{max} = 14$  s. Comparison between the value of *Commoners* predicted by the ABC-SMC method (grey), the *Supermodeling* (blue) and the *ground truth* (points), where lines are averaged predictions, while the boundaries of the shaded areas are at  $mean \pm one\ standard\ deviation$  for each *ground truth* point: **(Top)** actual; **(Bottom)** normalized to the value of the *ground truth*. (Color figure online)

schemes, the current studies usually introduce either small algorithmic nuances (i.e. [6, 15]) or algorithm implementations that support parallelization (i.e. [19]). For Kalman-filter-based data assimilation, the studies propose faster implementations of the algorithms [26] or hybridization with the ABC-SMC method (e.g. [8]). However, the aforementioned optimization approaches do not change the basic paradigms or improve DA performance radically.

In the era of deep learning, formal predictive models are often replaced (or supplemented) with faster data models for which the role of data assimilation in estimating parameters is played by the learning of black box (e.g., neural network) parameters. In general, learning a black box is a simpler procedure than data assimilation to a formal model. A very interesting data modeling concept, very competitive with formal models in the prediction of spatio-temporal patterns in chaotic systems, is that of Echo State Machines [16, 22], particularly the Reservoir Computing (RC) approach [23]. No prior model based on physics or other knowledge is used.

In contrast, the *Supermodeling* paradigm, unlike the purely data-based RC and DA approaches, relies on the knowledge already encoded in formal models and on the partial synchronization of the chosen imprecise *sub-models* to



**Fig. 4.** The results for densely sampled data and  $t_{max} = 14$  s. Comparison between the value of *Elites* predicted by the ABC-SMC method (grey), the *Supermodeling* (green) and the *ground truth* (points), plotted as in Fig. 3 (Color figure online)

supplement the knowledge contained in any one sub-model. The original type of supermodel relied on synchronization of the *sub-models* by nudging them to one another, while simultaneously *nudging* them to the GT data [4]. The inter-model nudging effectively gives inter-model data assimilation, with nudging coefficients that can be estimated based on overall error relative to truth. Thus standard DA methods, having been employed first to estimate states, then to adjust a model itself by estimating its parameters, are now used to estimate inter-model couplings in a suite of models - an even higher level of abstraction in the application of DA [9,10]. This type of *Supermodeling* was successfully used for ensembling toy dynamical models [4,10] like *Lorenz systems* (Lorenz 63, Lorenz 84) and for combining simplified climate models (see e.g., [37]).

Recent results showed that the *Supermodeling* approach can also be applied in modeling complex dynamical biological processes such as tumor evolution. In [13] we demonstrated that in a *Supermodel* of melanoma the tumor evolution can be controlled by the *sub-models'* coupling factors  $\mathbf{C}$ , producing a few qualitatively different tumor evolution patterns observed in reality. Recently, we have successfully assimilated *ground truth* data to the supermodel, using genetic algorithms [33]. However, due to computational complexity and the need for heavy High-Performance Computing, we are now implementing the more efficient procedure described in this paper.

## 5 Conclusions and Future Work

Herein we propose a novel metaprocedure for computational modeling, rooted in an extended use of data assimilation. It leads to a radical decrease in the number of free parameters, as compared to those in the source dynamical model, by ensembling a few imperfect *sub-models* - i.e., inaccurate and weak solutions of a classical DA-based pre-training scheme – within a single *Supermodel*. The case study demonstrates that due to the *sub-models*' synchronization, a small number of the *Supermodel* metaparameters can be estimated, based on assimilated observations, much faster than the full set of parameters in the overparametrized source model. Consequently, “effective parameter estimation” based on *Supermodeling* can produce more accurate predictions than those that could be obtained using traditional data assimilation methods to estimate a single model's parameters in reasonable time. It is crucial to mention that DA-based *Supermodeling* can be used with any given data assimilation procedure. The ABC-SMC algorithm was used here as the baseline classical DA method. *Supermodeling* plays only the role of the meta-framework dedicated to accelerating the modeling process.

We realize that our results can be treated as preliminary. A specific model was considered, and data assimilation was run on optimally selected working regimes and synthetic data. However, taking into account previous experience and more complicated phenomena simulated successfully by *Supermodeling*, one can expect that this procedure has wider prospects. Of course, there are still many unresolved issues, for example: how to generate efficiently the best *sub-models* and how many? How robust is the *Supermodel* against variations in noise, uncertainty and number of data samples? Herein we have assumed that the *sub-models* were generated in parallel because the pretraining of each can be performed independently. However, the total CPU time still increases proportionally with the number of *sub-models*. One can imagine that the *sub-models* could instead be generated by a single ABC-SMC procedure during the pre-training phase, by selecting more than one of the best solutions along the way. We plan to check this strategy in the very near future. We have taken as the *ground truth* the exact results from the reference (baseline) model. It would be worthwhile to check the quality of *Supermodel* predictions for disturbed data, which better simulate real observations. We are also considering a case study where the *sub-models* are simplified versions of the baseline model (preliminary results can be found in [31]). This way, the differences between the *Supermodel* and the *ground-truth* simulator, could better reflect the differences between the computational model and reality. Summarizing, the application of *Supermodeling* can be an effective remedy to the *curse of dimensionality* problem, caused by model overparameterization.

**Acknowledgments.** W.D. and G.S.D. are thankful for support from the National Science Centre, Poland grant no. 2016/21/B/ST6/01539 and the funds assigned to AGH University of Science and Technology by the Polish Ministry of Science and Higher Education. G.S.D. was also supported by the ERC (Grant No. 648982). This research was supported in part by PLGrid Infrastructure.

## References

1. Andrieu, C., De Freitas, N., Doucet, A., Jordan, M.I.: An introduction to MCMC for machine learning. *Mach. Learn.* **50**(1–2), 5–43 (2003)
2. Asch, M., Bocquet, M., Nodet, M.: *Data Assimilation: Methods, Algorithms, and Applications*, vol. 11. SIAM, New Delhi (2016)
3. Bain, A., Crisan, D.: *Fundamentals of Stochastic Filtering*, vol. 60. Springer, New York (2008). <https://doi.org/10.1007/978-0-387-76896-0>
4. Van den Berge, L., Selten, F., Wiegerinck, W., Duane, G.: A multi-model ensemble method that combines imperfect models through learning. *Earth Syst. Dynam. Discuss* **1**, 247–296 (2010)
5. Bergemann, K., Reich, S.: A mollified ensemble kalman filter. *Q. J. R. Meteorol. Soc.* **136**(651), 1636–1643 (2010)
6. Clarté, G., Robert, C.P., Ryder, R., Stoehr, J.: Component-wise approximate Bayesian computation via gibbs-like steps. *arXiv preprint [arXiv:1905.13599](https://arxiv.org/abs/1905.13599)* (2019)
7. Del Moral, P., Doucet, A., Jasra, A.: An adaptive sequential monte carlo method for approximate bayesian computation. *Stat. Comput.* **22**(5), 1009–1020 (2012)
8. Drovandi, C., Everitt, R.G., Golightly, A., Prangle, D.: Ensemble MCMC: accelerating pseudo-marginal MCMC for state space models using the ensemble kalman filter. *arXiv preprint [arXiv:1906.02014](https://arxiv.org/abs/1906.02014)* (2019)
9. Duane, G.: Data assimilation as artificial perception and supermodeling as artificial consciousness, pp. 209–222 (2012)
10. Duane, G.: Synchronicity from synchronized chaos. *Entropy* **17**(3), 1701–1733 (2015)
11. Duane, G., Grabow, C., Selten, F., Ghil, M.: Introduction to focus issue: synchronization in large networks and continuous media - data, models, and supermodels. *Chaos: Interdisc. J. Nonlinear Sci.* **27**(12), 126601 (2017)
12. Dzwiniel, W., Klusek, A., Paszyński, M.: A concept of a prognostic system for personalized anti-tumor therapy based on supermodeling. *Procedia Comput. Sci.* **108**, 1832–1841 (2017)
13. Dzwiniel, W., Klusek, A., Vasilyev, O.V.: Supermodeling in simulation of melanoma progression. *Procedia Comput. Sci.* **80**, 999–1010 (2016)
14. Fearnhead, P., Prangle, D.: Constructing summary statistics for approximate bayesian computation: semi-automatic approximate bayesian computation. *J. R. Stat. Soc.: Ser. B (Stat. Methodol.)* **74**(3), 419–474 (2012)
15. Ionides, E.L., Nguyen, D., Atchadé, Y., Stoev, S., King, A.A.: Inference for dynamic and latent variable models via iterated, perturbed bayes maps. *Proc. Natl. Acad. Sci.* **112**(3), 719–724 (2015)
16. Jaeger, H.: Echo state network. *Scholarpedia* **2**(9), 2330 (2007)
17. Järvenpää, M., Gutmann, M.U., Pleska, A., Vehtari, A., Marttinen, P., et al.: Efficient acquisition rules for model-based approximate bayesian computation. *Bayesian Anal.* **14**(2), 595–622 (2019)
18. Kalnay, E.: *Atmospheric Modeling, Data Assimilation and Predictability*. Cambridge University Press, Cambridge (2003)
19. Klinger, E., Rickert, D., Hasenauer, J.: pyABC: distributed, likelihood-free inference. *Bioinformatics* **34**(20), 3591–3593 (2018)
20. Kypriaios, T., Neal, P., Prangle, D.: A tutorial introduction to bayesian inference for stochastic epidemic models using approximate bayesian computation. *Math. Biosci.* **287**, 42–53 (2017)

21. Klusek, A., Łoś, M., Paszynski, M., Dzwiniel, W.: Efficient model of tumor dynamics simulated in multi-GPU environment. *Int. J. High Perform. Comput. Appl.* **33**, 489–506 (2018). <https://doi.org/10.1177/1094342018816772>
22. Lukoševičius, M.: A practical guide to applying echo state networks. In: Montavon, G., Orr, G.B., Müller, K.-R. (eds.) *Neural Networks: Tricks of the Trade*. LNCS, vol. 7700, pp. 659–686. Springer, Heidelberg (2012). [https://doi.org/10.1007/978-3-642-35289-8\\_36](https://doi.org/10.1007/978-3-642-35289-8_36)
23. Lukoševičius, M., Jaeger, H.: Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.* **3**(3), 127–149 (2009)
24. Marjoram, P., Molitor, J., Plagnol, V., Tavaré, S.: Markov chain monte carlo without likelihoods. *Proc. Natl. Acad. Sci.* **100**(26), 15324–15328 (2003)
25. Motesharrei, S., Rivas, J., Kalnay, E.: Human and nature dynamics (handy): modeling inequality and use of resources in the collapse or sustainability of societies. *Ecol. Econ.* **101**, 90–102 (2014)
26. Reich, S.: Data assimilation: the schrödinger perspective. *Acta Numerica* **28**, 635–711 (2019). <https://doi.org/10.1017/S0962492919000011>
27. Saltelli, A., Annoni, P., Azzini, I., Campolongo, F., Ratto, M., Tarantola, S.: Variance based sensitivity analysis of model output. Design and estimator for the total sensitivity index. *Comput. Phys. Commun.* **181**(2), 259–270 (2010)
28. Särkkä, S.: *Bayesian Filtering and Smoothing*, vol. 3. Cambridge University Press, Cambridge (2013)
29. Selten, F.M., Duane, G., Wiegerinck, W., Keenlyside, N., Kurths, J., Kocarev, L.: Supermodeling by combining imperfect models. *Procedia Comput. Sci.* **7**, 261–263 (2011)
30. Selten, F.M., Schevenhoven, F.J., Duane, G.S.: Simulating climate with a synchronization-based supermodel. *Chaos: Interdisc. J. Nonlinear Sci.* **27**(12), 126903 (2017)
31. Sendera, M.: Data adaptation in handy economy-ideology model. arXiv preprint [arXiv:1904.04309](https://arxiv.org/abs/1904.04309) (2019)
32. Shen, M.L., Keenlyside, N., Bhatt, B., Duane, G.: Role of atmosphere-ocean interactions in super-modeling the tropical pacific climate. *Chaos: Interdisc. J. Nonlinear Sci.* **27**(12), 126704 (2017)
33. Siwik, L., Los, M., Klusek, A., Pingali, K., Dzwiniel, W., Paszynski, M.: Supermodeling of tumor dynamics with parallel isogeometric analysis solver. arXiv preprint [arXiv:1912.12836](https://arxiv.org/abs/1912.12836) (2019)
34. Sobol, I.M.: Global sensitivity indices for nonlinear mathematical models and their Monte Carlo estimates. *Math. Comput. Simul.* **55**(1–3), 271–280 (2001)
35. Sunnåker, M., Busetto, A.G., Numminen, E., Corander, J., Foll, M., Dessimoz, C.: Approximate bayesian computation. *PLoS Comput. Biol.* **9**(1), e1002803 (2013)
36. Toni, T., Stumpf, M.P.: Simulation-based model selection for dynamical systems in systems and population biology. *Bioinformatics* **26**(1), 104–110 (2009)
37. Wiegerinck, W., Selten, F.: Attractor learning in synchronized chaotic systems in the presence of unresolved scales. *Chaos: Interdisc. J. Nonlinear Sci.* **27**(12), 126901 (2017)
38. Wiegerinck, W., Burgers, W., Selten, F.: On the limit of large couplings and weighted averaged dynamics. In: Kocarev, L. (ed.) *Consensus and Synchronization in Complex Networks*. UCS, pp. 257–275. Springer, Heidelberg (2013)