Evaluation of Kalman Filter for meteorological time series imputation for Eddy Covariance applications

Simone Massaro

First Supervisor: Dr. Franziska Koebsch
Second Supervisor: Prof. Dr. Fabian Sinz

Master's Thesis
at the Faculty of Forest Science and Forest Ecology
of the Georg-August-Universität Göttingen

Master's degree in Forest and Ecosystem Sciences
study track Ecosystem Analysis and Modelling

21 March 2023

Abstract

Eddy Covariance (EC) is a state-of-the-art technique to measure greenhouse gases exchanges. EC towers include measurement of meteorological variables, but due to instrument failures the data is not always available. Many use cases of EC data, especially land surface modelling, require continuous meteorological time series as input. Therefore, it is necessary to impute the gaps in the meteorological time series. ONEFlux, one of the most widely used EC post-processing pipeline, imputes the missing data using either Marginal Distribution Sampling (MDS), which uses other observations from similar meteorological conditions, or ERA-Interim (ERA-I), which is a global meteorological dataset. The imputation performance of those methods is limited for short and medium gaps (up to 1 week), which constitute the majority of EC meteorological gaps. In this work, I assess an imputation method for meteorological variables based on a Kalman Filter (KF). It has the advantages of combining in the prediction information from the ERA-I dataset, inter-variable correlation and temporal autocorrelation. Moreover, the KF is a probabilistic method, so for each data point the prediction is not a single value but an entire distribution, which provides an interpretable quantification of uncertainty of the model predictions.

I evaluate the KF by comparing the imputation performance with the state-of-the-art approaches (MDS and ERA-I) using data from the FLUXNET 2015 site of Hainich (DE-Hai) with gaps up to one week long. The KF outperforms the state-of-the-art approaches across all analyzed variables, except for precipitation, for which all methods are comparable. I observed an average reduction of the imputation error of 33 % compared to ERA-I and 57 % compared to MDS, when excluding precipitation. I further explore aspects that influence the performance of the KF: in general the error increases with the gap length only up to 24 hours, the use of ERA-I data improves the model predictions and the inter-variable correlation is effectively utilized. The main limitations of KF approach are: 1) the best performance is achieved only when fine-tuning the model parameters to the specific conditions of the gap, which increases the deployment complexity; 2) the current implementation of the KF is affected by numerical stability issues, which in case all variables are missing limits the maximum gap length to 15 hours; 3) careful initialization of the KF parameters and selection of the training conditions are required to mitigate the difficulty in learning the models parameters. However, I expect that all those issues can be resolved or at least significantly mitigated by further research.

Contents

1 Introduction

1.1 State of the art

Eddy Covariance Eddy Covariance (EC) is a state-of-the-art technique for measuring greenhouse gases and energy exchange between ecosystems and the atmosphere [2]. The technique allows for non-destructive measurements at the ecosystem level with a high temporal resolution (30 minutes). EC data is used for ecological and physiological research of ecosystems, for example to estimate the relation of forest age and carbon balance [4] or the effects of extreme events [30]. In addition, EC data is a key element for the validation and calibration of ecosystem process models and remote sensing observations [37]. At its core, the EC technique utilizes a 3D anemometer and a gas analyzer, which allows estimating the fluxes of interests (e.g. CO_2, H_2O, CH_4). Beside the fluxes, an EC setup commonly comprises measurements of meteorological variables and ecosystem parameters. This additional data provides the context to use and interpret the fluxes measurements.

Gaps in meteorological variables The acquisition of meteorological variables can be interrupted by failures in the instruments or power outages, resulting in gaps in the time series [2]. The presence of meteorological gaps is a problem for several uses of the EC data. An important use case of EC is the validation of Land Surface Models (LSM) [3, 20, 8, 29], which are process based model that estimate fluxes and include meteorological conditions as inputs. The errors of Land Surface Models deriving from inaccuracies in the input are comparable to the errors arising from the limitation in the models' formulation [52]. Secondly, meteorological observations are used as a driver to impute gaps in the fluxes measurements [2], which in turn requires complete meteorological time series. Finally, if the observations are aggregated, for example to compute the weekly average of meteorological conditions, the missing data leads to inaccurate results.
The described use cases highlight the need for high quality continuous meteorological measurements, that reflect the condition at the EC station. Redundant instruments and power supply on the site, reduce the amount of missing data [2]. However, even a redundant system is subject to failures, hence statistical models are used for imputing the remaining gaps [2].

Imputation approaches In general there are three approaches to obtain information on missing values in a multivariate time series and thus impute gaps: 1) use other observations of the missing variable to make predictions about the gap, in particular the variable *temporal autocorrelation* can be exploited to reconstruct the missing data [34]; 2) use *statistical dependency* between variables, if not all variables are missing then the dependency between variables can be used for imputing the missing variable [32]; 3) use *other independent measurements*, if another compatible and continuous time series are available they can be used for imputation [49]. For instance, in the case of EC meteorological variables, an independent time series can be obtained from a nearby meteorological station or a weather model reanalysis.
Imputation of missing values has been extensively researched and a wide range of methods have been developed, ranging from simply replacing with the mean to more advanced approaches employing deep neural networks [35, 19, 9, 16, 51, 10]. There exist several methods specifically developed to impute meteorological time series [13, 26]. However, those methods cannot be directly employed for the imputation of EC data, as it has some specific characteristics: the absence of a spatial component (i.e. EC sites are too distant from each other), the high temporal resolution and the relatively high number of variables.

Current imputation methods EC community EC post-processing pipelines impute meteorological time series. Arguably the most widely used post-processing pipeline is ONEFlux [39], which is adopted by several large networks such as FLUXNET, the global EC network, ICOS the European network as well as AmeriFlux, the American EC network. ONEFlux uses two different methods for imputing the meteorological data: Marginal Distribution Sampling (MDS) and ERA-Interim (ERA-I). The final gap-filled meteorological product uses either MDS or ERA-I, depending on the quality flag from MDS.

MDS Marginal Distribution Sampling [43] imputes the missing value by using the average of all the other data points observed in similar conditions. The similarity is both temporal, only observations from a limited time window around the gap are considered, and meteorological, the data points are restricted to times when other variables are similar. The algorithm selects all the data points where the value of the driver variables (other meteorological variables) is within a fixed threshold of the values observed at the missing data point. All the observations of the variable of interest from the selected data points are then averaged to generate the filling value. MDS starts with a time window of 7 days and if no similar conditions are found in this time frame the window is progressively increased. If a driver variable is also missing, it is not used in the selection of similar conditions. In case no similar conditions are found in a time window of 14 days or all drivers are missing, the MDS fails and the imputation is done using the average value at the same time of the day. For gaps longer than 140 days the method cannot impute the gap. MDS imputes each data point and each variable separately. It mainly uses statistical dependency between variables and in a limited way the variable's temporal autocorrelation.
The algorithm implemented in ONEFlux uses as drivers the Incoming Shortwave Radiation (SW_IN), the Air Temperature (TA) and Vapor Pressure Deficit (VPD). If either TA or VPD is missing, SW_IN is used as the only driver. MDS has a quality flag with three possible values (i.e. 1,2,3) that depends on the size of the time window. In ONEFlux MDS is used only if the quality flag is 1, which in general means that similar conditions are found in a time window smaller than 14 days (see Figure A1 in Reichstein et al. [43] for details).

ERA-Interim ERA-Interim (ERA-I) is a global meteorological dataset provided by the European Centre for Medium-range Weather Forecast (ECMWF) [15]. Weather forecast models are used to reanalyze past observations and produce a continuous and complete meteorological dataset for the entire globe. The main drawback is the low spatial resolution and temporal resolution, that are respectively 80km and 3 hours. Moreover, only a subset of the meteorological variables are available in ERA-I (Table 2) and the data is not available in real-time but with a three months delay.
In order to use ERA-I in the EC context, the observations have to be temporally downscaled to match the half-hourly frequency of EC data. Furthermore, the performance of ERA-I imputation can be improved by removing the systematic bias for each site. Both steps are performed in ONEFlux as described in Vuichard and Papale [49]. The error correction is performed using a different linear regression for each site and each variable.
The accuracy of the ERA-I imputation is independent of the length of the gap. This is advantageous for long gaps, as ERA-I data includes long-term evolution of the weather, which is not possible to predict by only analyzing the local time series. At the same time, for short gaps, the local conditions can provide a more accurate prediction. In fact, ONEFlux imputes short gaps using MDS, while utilizes ERA-I for long gaps.

Other methods Beyond ONEFlux, there are several other established EC post-processing pipelines, which impute meteorological data. However, the imputation approaches in other libraries, like REddyProc [50] or OzFlux [25], are similar. REddyProc employs only MDS, while OzFlux uses both MDS and ERA-I. In addition, OzFlux also includes data from the Australian Weather Service (AWS) and for each gap it utilizes either ERA-I or AWS, depending on which dataset has the smallest error for a time window of 90 days around the gap.

1.2 Potential for improvement

The imputation quality of the current methods is sometimes limited (Figure 1). In particular, the imputation using MDS often results in relatively high error and unrealistic patterns. There are two possible directions to improve the accuracy of the imputation: make a better use of the variables temporal autocorrelation; combine the information from ERA-I data and analysis of the local time serie for every prediction, instead of using the two approaches independently. Moreover, the current methods do not provide a measure of the uncertainty of the predictions.

Temporal autocorrelation MDS uses the temporal autocorrelation only in a limited way, as it takes the average of the missing variable across the whole time window and does not weight the data depending on the proximity to the gap. Similarly, the bias correction in ERA-I uses the entire dataset from a site, thus more importance is not assigned to the conditions around the gap. This is a suboptimal use of available data, as the observations close to the gap have the highest correlation with the data in the gap and the meteorological variables have an overall high temporal autocorrelation (Figure 4a). This is particularly relevant for short and medium gaps (shorter than 1 week), which are the majority in the EC context. In FLUXNET 2015 [39], the most extensive EC dataset with over 200 sites, almost 99 % of gaps of meteorological variables are shorter than a week (appendix Figure A.8).

Combination of imputation approaches ONEFlux employs both ERA-I and MDS, but the two methods are used independently, not combined for each prediction. The criteria to select the method to use is only the MDS quality control flags. The information on the missing data from temporal autocorrelation, dependency with other variables and other source of measurements (e.g. ERA-I) can be combined to make a more accurate prediction.

Uncertainty A limitation of the current methods is the lack of a robust assessment of the uncertainty of the imputed values. MDS has a quality flag, but it is limited to only three possible values and it derives from constant thresholds. Moreover, in the final ONEFlux product, the quality flag indicates only which gap filling method was used. Ideally, each predicted data point has an associated uncertainty, which varies continuously and is interpretable, with the same physical unit of the variable. In this way, the level of confidence of the model in each prediction is available to the data user. The uncertainty can be either used to discard the data when it is above a custom threshold, which can change depending on the application, or directly included in the downstream calculations.

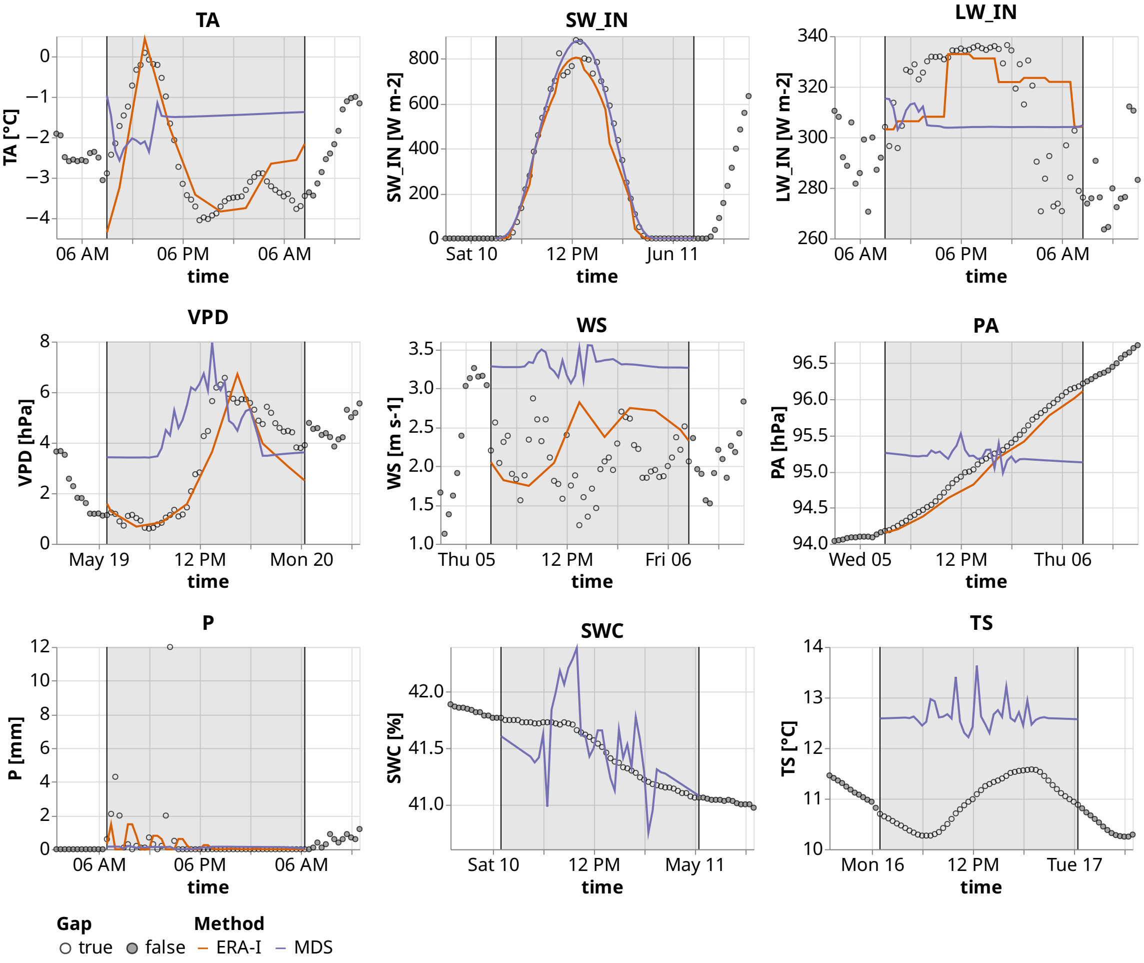


Figure 1: Time series to visualize imputation using state-of-the-art methods: ERA-Interim (ERA-I in orange) and Marginal Distribution Sampling (MDS in purple). For each variable, one gap 24 hours long was created, with only one variable missing. The gray shaded area and the vertical black lines delimit the artificial gaps, where the observations are not available to the model but are used to assess the imputation performance.

1.3 Contribution of this work

In this work, I focused on a method that combines all three imputation approaches and includes interpretable quantification of uncertainty. Probabilistic machine learning algorithms are particularly suited, as they directly provide uncertainty for the predictions. Gaussian Processes (GP) are one of the most important probabilistic algorithms [22]. GP can model interactions between all data points, for example they can consider both a yearly and a daily pattern in the data. This, however, is connected to their main drawback: the computation cost scales cubically with the number of observations, making the use of GP computationally prohibitive. To overcome this limitation, several approximations, such as sparse GP, have been developed. The Kalman Filter (KF) can be viewed as a special kind of GP, which models the time at discrete steps and where all the information about past and future observations is stored in a latent state. This drastically improves the computation efficiency, which scales linearly in the number of observations, but limits the ability to model processes with long time scales. However, in the context of EC meteorological imputation, this is an acceptable tradeoff as the majority of gaps are not long. Another advantage of the KF is the ability to include the ERA-I data in the predictions.
The aim of this work is to develop and test an imputation method for meteorological time series in the context of EC that employs a KF, as it promises more accurate predictions through a more efficient use of temporal autocorrelation in combination with the ERA-I data. Moreover, the KF provides a quantification of the predictions' uncertainty. The imputation performance of the KF is evaluated by comparing it with the state-of-the-art methods (i.e. ERA-I and MDS). Then the aspects that affect the performance of the KF are assessed: the impact of the length of the gap, the advantage of including ERA-I data, the importance of inter-variable correlation and different training scenarios. The data from the EC site of Hainich, Germany, is used to train and evaluate the model.

2 Methods

2.1 Kalman Filter theory

Kalman Filter (KF) models over time a latent variable x, that represents the state of the system. The state cannot be directly observed, but it is possible to observe meteorological variables y that reflect the state of the system. KF is a probabilistic machine learning algorithm, so it keeps track of the entire distribution of the latent variable [6]. The KF can update the state also when there are missing observations, hence the state is available for all time steps, which can in turn be used to predict the missing data points.
In order to model the state over time, assumptions on the behavior of the system are made. The first element is to model the time as a discrete variable. Then there are three key assumptions: 1) the states are connected by a Markov chain, which means that the state at time t depends only on the state at time $t-1$ and not on the states at previous times: $p(x_t|x_{t-1}) = p(x_t|x_{t-1}, x_{t-2}, \dots, x_0)$; 2) the value of the observed variables depends on the latent state; 3) all the relationships are linear and all distributions are Gaussian. Additionally, the mean of the state at time t may also depend on an external control variable c_t. This control variable does not depend on the state of the models, but provides information on the change of the state mean. Equations 1 and 2 describe the assumptions on the behavior of the system:

$$p(x_t|x_{t-1}) = \mathcal{N}(x_t; Ax_{t-1} + d + Bc_t, Q) \tag{1}$$

$$p(y_t|x_t) = \mathcal{N}(y_t; Hx_t + b, R) \tag{2}$$

The probability distributions of the state are computed using Bayesian inference. The computational cost of probabilistic inference is drastically reduced in this context, as can be performed using only linear algebra operations since all the relations are linear and all distributions are Gaussian.

The aim of the KF is to compute for every data point the probability distribution of the missing observations, y_t^g, given all the other observations, Y^{ng}, and the control variable, C: $p(\hat{y}_t^g \mid Y^{ng}, C)$. To achieve this, the KF recursively computes intermediate distributions (Figure 2) with the same set of operations repeated for every time t.

The first step is to compute the *predicted state*, x_t^-, that is obtained from the previous state, x_{t-1}, and the *control variable*, c_t. This represents the conditional distribution of the state, given all observations until time $t-1$, $Y_{0:t-1}^{ng}$, and the control variable until time t: $C_{0:t}$, $p(x_t \mid Y_{0:t-1}^{ng}, C_{0:t})$.

The second step is to update the predicted state using the *observations* at time t, y_t, to obtain the *filtered state*, x_t^f. Observations can be partially or totally missing. This is the conditional distribution of the state given all the observation until time t: $p(x_t \mid Y_{0:t}^{ng}, C_{0:t})$. These two steps make the filtering pass of the KF and are iteratively repeated for every time step in the observed dataset, starting from time 0. The last operation is to compute the smoothed state x_t^s, which is obtained by updating the filtered state using the information from the observations after time t. This computes the conditional distribution of the state given all the observations and control variables: $p(x_t \mid Y^{ng}, C)$. Finally, the distribution of the missing observations, $p(y_t^g \mid Y^{ng}, C)$, can be computed directly from the smoothed state for each time step.

The model always considers the entire probability distribution for the state, which is a Gaussian distribution, $p(x_t) = \mathcal{N}(x_t; m_t, P_t)$, so stores for each state at each time step the mean, m_t, and the covariance, P_t. Similarly, the model predictions are a multivariate Gaussian distribution $p(y_t \mid Y^{ng}, C) = \mathcal{N}(y_t; \mu_{y_t}, \Sigma_{y_t})$.

Time update The first step in a KF is computing the probability distribution of the predicted state x_t^-, from the filtered state at the previous time step x_{t-1}^f and the control variable c_t. The value of the control variable at time t affects the value of state mean, but does not influence the state covariance. The initial state of the system has the following distribution: $p(x_0) = \mathcal{N}(x_0; m_0, P_0)$. Using Equation 1 and the properties of a linear map of Gaussian distributions the following equation can be derived [6, 22]:

$$\begin{aligned} p(x_t \mid Y_{0:t-1}^{ng}, C_{0:t}) &= \mathcal{N}\left(x_t; m_t^-, P_t^-\right) \\ m_t^- &= Am_{t-1} + Bc_t + d \\ P_t^- &= AP_{t-1}A^\top + Q \end{aligned} \tag{3}$$

Measurement update The predicted state is updated to obtain the distribution of the filtered state, using the current observation y_t. Equation 2 describes the distribution of y_t given x_t. Using Bayes' theorem, it is possible to compute the distribution of x_t given an observation, y_t [6, 22]:

$$\begin{aligned} p(x_t \mid Y_{0:t}^{ng}, C_{0:t}) &= \mathcal{N}(x_t; m_t, P_t) \\ z_t &= Hm_t^- + d \\ S_t &= HP_t^- H^\top + R \\ K_t &= P_t^- H^\top S_t^{-1} \\ m_t &= m_t^- + K_t(y_t - z_t) \\ P_t &= (I - K_t H)P_t^- \end{aligned} \tag{4}$$

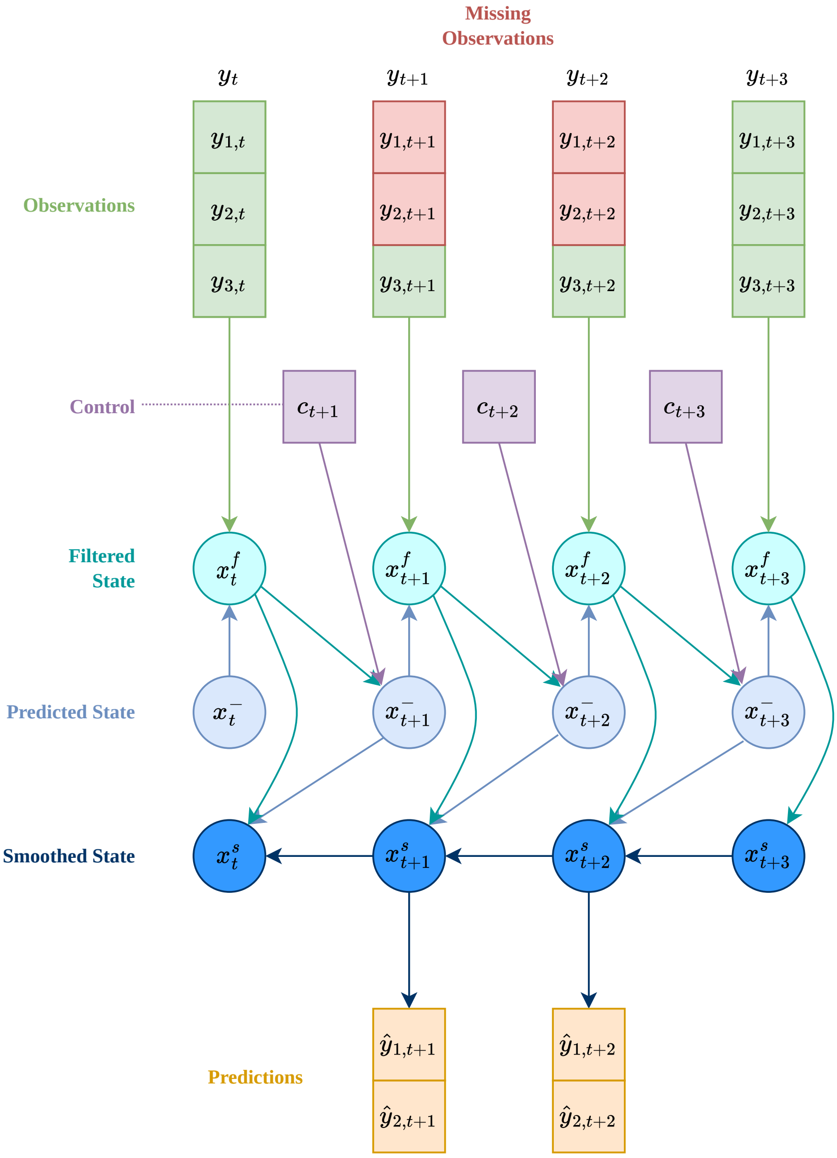


Figure 2: Schematic representation of an example Kalman Filter. The green squares represent the observations of a single variable at a specific time, the observations may be missing (red squares). The blue circles represent the latent state, specifically the three versions of the state modelled by the KF: filtered state (cyan), predicted state (light blue) and smoother state (dark blue). The control variables are shown in purple. All the arrows show a direct dependency for the computation of each element.

Missing observations The KF is able do deal with missing observations and can update the state even in that case. If all the observations at time t are missing, the measurement update step is skipped and the filtered, x_t^f, is the same of the predicted state, x_t^-. If only some observations in y_t are missing, then a partial measurement step is performed. The vector containing the observations that are not missing at time t, y_t^{ng}, can be expressed as a linear transformation of y_t:

$$y_t^{ng} = M_t^{ng} y_t \tag{5}$$

where M_t^{ng} is a mask matrix that is used to select the subset of y_t that is observed. $M_t^{ng} \in \mathbb{R}^{n^{ng} \times n}$, $y_t \in \mathbb{R}^n$, and $y_t^{ng} \in \mathbb{R}^{n_{ng}}$. The mask is made of rows which are all zeros, but for an entry 1 at the column corresponding to the of the index of the non-missing observation.
For example, if $y_t = [y_{0,t}, y_{1,t}, y_{2,t}]^\top$ and $y_{0,t}$ is the missing observation, then:

$$M_t^{ng} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Using the properties of linear projections of a Gaussian distribution, it is possible to derive the distribution $p(y_t^{ng} \mid x_t)$ from $p(y_t \mid x_t)$:

$$p(y_t^{ng}|x_t) = p(M_t^{ng} y_t|x_t) = \mathcal{N}\left(y_t^{ng}; M_t^{ng} H x_t + M_t^{ng} b, M_t^{ng} R (M_t^{ng})^\top\right)$$

Therefore, it is possible to perform the measurement update step when some observations are missing using a variation of Equation 4, where H is replaced by $M^{ng}H$, b by $M^{ng}b$ and R by $M^{ng}R(M^{ng})^\top$. In this way, H varies between each time step depending on which variables are missing.

Smoothing In the smoothing step, the filtered state at time t is updated using the observations after time t. A widely applied set of equations for the smoothing pass is the Rauch-Tung-Striebel Smoother [42]. They calculate the smoothed state, x_t^s, from the filtered state and time t, and the smoothed and filtered and at time $t+1$:

$$\begin{aligned} p(x_t \mid Y^{ng}, C) &= \mathcal{N}(x_t; m_t^s, P_t^s) \\ G_t &= P_t A^\top (P_{t+1}^-)^{-1} \\ m_t^s &= m_t + G_t(m_{t+1}^s - m_{t+1}^-) \\ P_t^s &= P_t + G_t(P_{t+1}^s - P_{t+1}^-) G_t^\top \end{aligned} \tag{6}$$

For the last time step, the smoothed state is set to be equal to the filtered state.

Predictions The predicted distribution of the observed variables in a gap, $\hat{y}_t^g$, can be obtained directly by the distribution of the smoothed state, $p(x_t|Y^{ng}, C)$. The missing observed variables a time t are $y_t^g = M_t^g y_t$, where M^g has a similar definition as M_t^{ng} in Equation 5, but selects the missing observations instead of non-missing data. The following equation can see derived (appendix Equation B.1):

$$\begin{aligned} p(\hat{y}_t^g \mid Y^{ng}, C) &= \mathcal{N}(\hat{y}_t^g; \mu_{y_t}, \Sigma_{y_t}) \\ \mu_{y_t} &= M_t^g H m_t + M_t^g b \\ \Sigma_{y_t} &= M_t^g R (M_t^g)^\top + M_t^g H P_t^s H^\top (M_t^g)^\top \end{aligned} \tag{7}$$

The output of the KF model is the distributions of $p(y_t^g)$ for all gaps in the observed variables.

2.2 Kalman Filter implementation

KF is a widely used algorithm and there are several python libraries that implement it (e.g. statsmodels, pykalman, filterpy). However, no KF library was identified which meets all the requirements for this work. It is necessary to support gaps, partial measurements updates, control variables and be a numerically stable implementation. Therefore, a custom library for KF was developed using the PyTorch library, which has the advantage of automatic differentiation, possibility to use GPUs and better integration with other Machine Learning methods.

2.2.1 Numerical stability

The naive implementation of the KF equations suffers from numerically stability issues [33, 14]. Numerical instability arises from the fact that digital computers store numbers with only limited precision, which also varies depending on the value of the number. This results in a loss of information, so that some operations may be incorrectly performed by a computer (e.g. summing a big number and a small number).

For KF the components that are most affect by numerical instability are the covariance matrices. To analyze the stability of the operations on these matrices it is relevant to consider the condition number for inversion [33, 27], which provides an indication if the matrix is going to be singular on the numerical representation in the computer. The condition number $k(A)$ is the ratio between the biggest singular value and the smallest. The singular value is $\sigma^2(A) = \lambda(AA^\top)$, with $\lambda(A)$ being the eigenvalue of A:

$$k(A) = \frac{\sigma_{max}(A)}{\sigma_{min}(A)}$$

The condition number is 1 for well-conditioned matrices, and tends to infinite for ill-conditioned matrices. As a general rule, a matrix cannot be inverted when the reciprocal of the condition number for inversion is close to the machine precision $1/k(A) < \varepsilon$ [33].

Mitigation strategies The simplest approach to improve the numerical stability is to use higher accuracy in the representation of numbers [14]. Practically, this means to use 64 bit floats instead of 32 bit floats, which is the default in PyTorch.

Another way to improve the numerical stability is to reduce the condition number of the state covariance, P. A positive definite matrix has a square root factor, $P^{1/2}$, such as that $P = P^{1/2}(P^{1/2})^\top = P^{1/2}P^{\top/2}$. The Cholesky decomposition is an algorithm to find a square root of a matrix, however the Cholesky decomposition calculates only one of possibly many square roots of the matrix.

Utilizing $P^{1/2}$ instead of P doubles the effective numerical resolution of the KF [27, 14, 44]. This is due to the fact that the eigenvalues of $P^{1/2}$ are the square root of the eigenvalues of P, $\lambda(P) = \lambda^2(P^{1/2})$, thus the conditioning number of P is the square of the conditioning number of $P^{1/2}$. Therefore, if in the KF implementation P is never explicitly computed, the numerical stability of the KF is significantly improved. There are several implementations of a KF that follow this approach [41, 11, 5] and are generally called "square root" filter.

Implementation in PyTorch There are different approaches to square root filtering. According to Mohinder S. Grewal and Angus P. Andrews [33] the best approach is the UD-Filter [5], since it has the smallest computational cost. However, the filter is based on the UD factorization and a custom matrix factorization [33] and both of those algorithms cannot be efficiently implemented in PyTorch. The PyTorch function `torch.linalg.ldl_factor` performs a UD factorization, but it is an experimental function and is not differentiable. Moreover, the custom

matrix factorization would need to be implemented using scalar operations, which are not efficient with PyTorch eager execution.
For this reason, a square root filter that propagates square roots of the covariance matrices is implemented. In this way, all the required computations can be expressed in QR factorization, which is a numerically stable routine and is implemented in PyTorch.

Time update Square Root Filter From the equations of the time update step (Equation 3), it is possible to derive an algorithm to obtain $P_t^{1/2}$ given $P_{t-1}^{1/2}$ [33, eq. 6.60]. Defining:

$$W = \begin{bmatrix} AP_{t-1}^{1/2} & Q^{1/2} \end{bmatrix}$$

from Equation 3 the following is true:

$$\begin{aligned} WW^\top &= \begin{bmatrix} AP_{t-1}^{1/2} & Q^{1/2} \end{bmatrix} \begin{bmatrix} P_{t-1}^{\top/2} A^\top \\ Q^{\top/2} \end{bmatrix} \\ &= AP_{t-1}^{1/2} P_{t-1}^{\top/2} A^\top + Q^{1/2} Q^{\top/2} = AP_{t-1} A^\top + Q \\ &= P_t \end{aligned}$$

The next step is to factorize $W = LU$, where L is a lower triangular matrix and U is an orthogonal matrix, such as that $UU^\top = I$. Then $WW^\top = LU(LU)^\top = LUU^\top L^\top = LL^\top = P_t$. Hence, L is a square root of P_t. This procedure never explicitly computes P_t and requires only the factorization of a matrix, which is implemented efficiently and in a numerical stable way in the PyTorch `torch.linalg.qr` function. PyTorch does not support natively a LU decomposition, but it implements the QR factorization: $W = QR$, where Q is an orthogonal matrix and R an upper triangular matrix. This can be used to compute the LU factorization by performing a QR factorization of $W^\top$ and defining $L = R^\top$, as $W = (W^\top)^\top = (QR)^\top = R^\top Q^\top = LU$. The steps of the Square Root time update are:

1. let $W = \begin{bmatrix} AP_{t-1}^{1/2} & Q^{1/2} \end{bmatrix}$
2. do the LU factorization $W = LU$
3. set $P_t^{1/2} = L$

Measurement update Square Root Filter A similar procedure can be followed for the measurement update step of the filter [14]. The starting point is Equation 4, for simplicity the time subscripts are omitted in the following equations. Defining:

$$M = \begin{bmatrix} R^{1/2} & H(P^-)^{1/2} \\ 0 & (P^-)^{1/2} \end{bmatrix} \quad V = \begin{bmatrix} S^{1/2} & 0 \\ \bar{K} & P^{1/2} \end{bmatrix} \quad \bar{K} = KS^{1/2}$$

then $MM^\top = VV^\top$ (appendix Equation B.2). Therefore, by decomposing $M = LU$, then $MM^\top = LL^\top = VV^\top$ and the bottom right block of size $k \times k$ of L is a square root of P, where k is the number of dimensions of the state, $x_t \in \mathbb{R}^k$. The steps of the Square Root measurement update are:

1. let $M = \begin{bmatrix} R^{1/2} & H(P^-)^{1/2} \\ 0 & (P^-)^{1/2} \end{bmatrix}$

2. do the LU factorization of $M = LU$
3. $P^{1/2}$ is the bottom right $k \times k$ block of L

Predictions Square Root Filter The prediction equations for the square root filter are similar to the equations for the time update. Defining:

$$W = \begin{bmatrix} HP_t^{1/2} & R^{1/2} \end{bmatrix}$$

from Equation 7 the following is true:

$$\begin{aligned} WW^\top &= \begin{bmatrix} HP_t^{1/2} & R^{1/2} \end{bmatrix} \begin{bmatrix} P_t^{T/2}H^\top & R^{\top/2} \end{bmatrix} \\ &= HP_t^{1/2}P_t^{\top/2}H^\top + R^{1/2}R^{\top/2} = HP_tH^\top + R \\ &= \Sigma_{y_t} \end{aligned}$$

The steps of the Square Root predictions are:

1. let $W = \begin{bmatrix} HP_t^{1/2} & R^{1/2} \end{bmatrix}$
2. do the LU factorization of $W = LU$
3. set $\Sigma_{y_t}^{1/2} = L^\top$

Smoothing Square Root Filter The available literature for implementing a square root smoother is scarce compared to a square root filter. Therefore no solution has been identified to implement a square root smoother and a standard smoother is employed. Nonetheless, steps were taken to improve the numerical stability of the smoother. The computation in the smoother that is most numerically unstable is the inversion of P_{t+1}^- in Equation 6 [33]. The matrix inversion is avoided by using the `torch.cholesky_solve` function. It solves for X the linear system $P_{t+1}^- X = P_t A$, which is equivalent of computing $X = (P_t A^\top (P_{t+1}^-)^{-1})^\top$. This uses directly the square root $(P_{t+1}^-)^{1/2}$ to avoid the computation of P_{t+1}^-. A further step to improve the numerical stability is forcing the covariance matrix to be symmetric, by averaging to upper and lower part at after every time step, $P_{t,sym}^s = (P_t^s + (P_t^s)^\top)/2$, as suggested in Dan Simon [14]. This approach to numerical stability in the smoother is the same as applied by the statsmodels library [47].

2.3 Kalman Filter model

Table 1: Parameters of the Kalman Filter Model. n is the number of dimension of the observations, $k = 2n$ the number of dimensions of the state, n_{ctr} the number of dimensions of the control variable.

Parameter name	Notation	Shape	Initial value
State transition matrix	A	$k \times k$	$\begin{bmatrix} I & I \\ 0 & I \end{bmatrix}$
Observation matrix	H	$n \times k$	$\begin{bmatrix} I & 0 \end{bmatrix}$
State transition covariance	Q	$k \times k$	diag(0.1)
Observation covariance	R	$n \times n$	diag(0.01)
State transition offset	d	k	0
Observation offset	b	n	0
Control matrix	B	$k \times n_{ctr}$	$\begin{bmatrix} -I & I \\ 0 & 0 \end{bmatrix}$
Initial state mean	m_0	k	0
Initial state covariance	P_0	$k \times k$	diag(3)

Parameters The KF is implemented as PyTorch module, whose parameters are described in Table 1. There is no change over time of the parameters, and the state of the KF is initialized always at the same value from the parameters m_0 and P_0.

An important aspect for implementing a KF in PyTorch is constraining the parameters that represent covariance (Q, R and P_0) to be positive definite [6]. To achieve this goal, the optimizer works on a raw parameter, which is then transformed into a positive definite matrix. The transformation into a positive definite matrix is done by transforming the raw parameter into a lower triangular matrix with a positive diagonal. The diagonal is enforced to be positive by transforming the diagonal of the raw parameter with the softplus function, $x = \log(1+e^x)$, which is a positive function. In addition, a small positive offset, 1×10^{-5}, is added to the diagonal in order to avoid that the diagonal is close to zero, which may result in a positive semi-definite matrix. This implementation of the positive definite constraint makes it straightforward to obtain the Cholesky factor of the parameters, which are needed by the Square Root Filter, and at the same time the full parameters, which are needed by the smoother.

Parameters initialization The model parameters could be initialized using random values, however this would increase numerical stability issues and increase the training time. Moreover, if the initial parameters are very distant from the optimal ones, it is more likely for the optimization algorithm to find only a local minimum. The simplicity of the KF and the interpretability of its parameters allows to manually initialize the parameters with realistic values.

The parameters are initialized using a local linear trend model [18]. It assumes that the state of the system, x_t, is made by two components, the level x_{l_t} and the slope, x_{s_t}: $x_t = [x_{l_t}, x_{s_t}]^\top$. The observed variable, y_t, is equal to the state level with an addition of Gaussian white noise, ε_t; the state level is the sum of the previous state levels, $x_{l_{t-1}}$. The previous state slope, $x_{l_{t-1}}$, and a Gaussian white noise component, ν_{t-1}. The state slope is equal to the previous slope, $x_{s_{t-1}}$, and another Gaussian white noise component, ξ_{t-1}:

$$\begin{aligned} y_t &= x_{l_t} + \varepsilon_t & p(\varepsilon_t) &= \mathcal{N}(\varepsilon; 0, R) \\ x_{l_t} &= x_{l_{t-1}} + x_{s_{t-1}} + \nu_{t-1} & p(\nu_{t-1}) &= \mathcal{N}(\nu; 0, Q_\nu) \\ x_{s_t} &= x_{s_{t-1}} + \xi_{t-1} & p(\xi_{t-1}) &= \mathcal{N}(\xi; 0, Q_\xi) \end{aligned}$$

A KF can be used to model a system described above by selecting suitable parameters:

$$A = \begin{bmatrix} I & I \\ 0 & I \end{bmatrix} \quad Q = \begin{bmatrix} Q_\varepsilon & 0 \\ 0 & Q_\nu \end{bmatrix} \quad H = \begin{bmatrix} I & 0 \end{bmatrix}$$

The use of the slope component allows the KF to retain information about several previous and following observations, while a KF with only a level compotent would be limited to use only the obsevation one time step before or after the gap.

The local slope model can be extended by adding the state and observations offsets d and b and a control variable similarly to Equations 2 and 1. In the context of this work, the ERA-I dataset is used for the control variable. The control variable should quantify the change between consecutive states levels, so the difference between consecutive ERA-I observations is used. This can be achieved by setting the control variables to $c_t = [\text{ERA-I}_t, \text{ERA-I}_{t-1}]^\top$ and uses the control matrix to compute the difference between the ERA-I observations. Moreover, the control variable should have no effect on the state slope. An issue with this approach arises from the fact that some variables are not present in the ERA-I (i.e. $n_{ctr} < n$), thus there is not a one to one correspondence between the control variable and the observed variable, and hence the state level. This is solved by sorting the variables in a way that there is correspondence between the first n_{ctr} observed variables and the ERA-I variables. For those reason, the control matrix is initialized in such a way that the control variable influences only the level of the first n_{ctr} dimensions of the state level:

$$B = \begin{bmatrix} -I & I \\ 0 & 0 \end{bmatrix} \quad B \in \mathbb{R}^{k \times n_{ctr}} \quad I \in \mathbb{R}^{n_{ctr} \times n_{ctr}}$$

The state transition covariance Q and then observation covariance R are initialized as diagonal matrix with values of 0.1 and 0.01 respectively. These numbers have been chosen to represent an uncertainty in the state transition that is compatible with the standard deviation of the variables (i.e $\sigma = 1$ after standarization) and a low uncertainty in the observations.

The observation and state transition offsets are initialized to zero. The initial state is set to have as mean zero and as covariance diag(3). The number 3 is an arbitrary number bigger than the state transition covariance, which represents an high level of uncertainty for the initial state.

Loss Function The loss function used to train the model is the negative log likelihood, computed for each data point. At each time step, the model predicts a multivariate normal distribution $p(\hat{y}^g_t \mid Y^{ng}, C)$, which is used to compute the negative log likelihood given the actual observations y^g_t. The negative log likelihoods between different time steps in the same gap are summed, while the negative log likelihood is averaged between batches. The actual loss function of the model should be the log likelihood of the joint distribution $p(Y^g|Y^{ng}, C)$. However, the analytical form of the joint distribution cannot be easily derived from the KF equations. The log likelihood of marginal distributions is instead used, as it is a lower bound to the log likelihood of the joint distribution. Defining: $q(x)$ the predicted joint distribution, $p(x)$ the real joint distribution and $q_i(x)$ the marginal distribution at the ith time step:

$$q_i(x) = \int q(x_1, ..., x_k) dx_{\neg i}$$

If the family of distribution of $q(x \mid \theta)$, is the same of $\prod_i q_i(x \mid \theta)$, where θ are the model parameters; then:

$$\max_\theta \langle \log q(x \mid \theta) \rangle_{x \sim p(x)} \geq \max_\theta \langle \log \prod_i q_i(x \mid \theta) \rangle_{x \sim p(x)} \tag{8}$$

because $\prod_i q_i(x)$ is more restricted. This means that $q(x)$ fits at least as good as $\prod_i q_i(x)$. For the KF $q_i(x)$ is a Gaussian distribution, so $\prod_i q_i(x)$ is also a Gaussian distribution and Equation 8 holds.

Metrics The main metric used to assess the model performance is the *Root Mean Square Error* (RMSE):

$$\text{RMSE} = \sqrt{\frac{\sum_i^n (y_i^g - \hat{y}_i^g)^2}{n}}$$

The advantage of the RMSE is that it can also be used for non-probabilistic methods (e.g. MDS) and that its value has the same physical dimension as the observed variable. The main drawback is that it cannot be used for comparison between variables. For that, the *Standardized* RMSE is used, which is the RMSE computed on the standardized variables (Equation 9). It can be computed by transforming the RMSE (appendix Equation B.3):

$$\text{RMSE}_{\text{stand}} = \frac{\text{RMSE}}{\sigma_Y}$$

Other metrics, like the R^2 score and the Mean Absolute Percentage Error (MAPE) were evaluated, however none of them are suitable for this application. The R^2 is defined as $R^2 = 1 - (\sum_i^n (y_i - \hat{y}_i)^2)/(\sum_i^n (y_i - \bar{y})^2)$, if the denominator is close to zero, then value of R^2 tends to $-\infty$. Since the gaps are often short and several variables are constant over short periods (e.g. `SW_IN`, `SWC`) the denominator of the R^2 would be close to zero and the metrics cannot be effectively used. The mean absolute percentage error is defined as $\text{MAPE} = \frac{1}{n} \sum_{i=0}^{n-1} (|y_i - \hat{y}_i|)/(|y_i|)$, which tends to ∞ when y_i tends to 0, as zero is a possible value of several variables (e.g. `SW_IN`, `TA`) this metric cannot be employed. It would be possible to use R^2 or MAPE for a subset of variables and gap lengths, but this limits the ability to perform comparisons across different settings.

Performance considerations The iterative nature of the KF, where the current state depends on the previous state, makes it impossible to use PyTorch vectorization across different time steps. This can significantly limit the performance of the KF, especially when executed on GPUs. In order to mitigate this issue, all functions in the KF library support batches, so at every time step different data is processed in parallel.

2.4 Data

Data source The data used to evaluate the performance of the KF is from the Hainich (Germany) site. The EC site in Hainich (DE-Hai) is on a deciduous beech forest and it is managed by the bioclimatology department at the University of Göttingen. The source of the data is the FLUXNET 2015 Dataset [39], which for Hainich includes measurements with a 30 minutes frequency between 2000 and 2012. In total 227952 observations are available. For simplicity, the entire dataset was used for the model training, which includes also gap-filled observations. All the meteorological variables that are gap-filled in the FLUXNET 2015 dataset were selected for the analysis (Table 2).

The FLUXNET 2015 dataset also includes the ERA-I dataset for each site. The data is bias corrected and temporally downscaled. The ERA-I data is used as the control variable for the KF. All the variables of interest are present in ERA-I, except for `TS` and `SWC`.

Table 2: Meteorological variables used to evaluate the Kalman Filter imputation. ERA-I column indicates whether the variable is available in the ERA-Interim dataset.

Variable mame	**Abbreviation**	**Unit**	**ERA-I**
Air Temperature	**TA**	°C	✓
Incoming Shortwave Radiation	**SW_IN**	W/m^2	✓
Incoming Longwave Radiation	**LW_IN**	W/m^2	✓
Vapour Pressure Deficit	**VPD**	hPa	✓
Wind Speed	**WS**	m/s	✓
Air Pressure	**PA**	hPa	✓
Precipitation	**P**	mm	✓
Soil Temperature	**TS**	°C	✗
Soil Water Content	**SWC**	%	✗

Data preparation pipeline The dataset needs to be pre-processed by dividing it into data blocks, adding an artificial gap and then standardize. The data preparation pipeline takes as input a list of items and outputs the data in a format suitable for training. Each item provides all the information about a gap with the following fields: a) `i` the index of the block; b) `shift` the offset of the data block; c) `var_sel` the variables missing in the gap; d) `gap_len` the gap length. The pipeline performs the following steps: 1) splits the index of complete data frame from Hainich into blocks of a given length and selects the `i`$^{\text{th}}$ element; 2) adds the shift to move the starting point of the data block and select the data from the data frame. For ERA-I it also adds the observations with a lag 1, so that at the time t the model has access to the ERA-I observations both at time t and $t-1$; 3) creates one continuous artificial gap in the middle of the block for the variables specified in `var_sel` and with a length of `gap_len` 4) converts from Pandas data frame to a PyTorch tensor 5) standardizes each variable, using the mean, μ_Y, and standard deviation, σ_Y, of the whole dataset:

$$y_t^z = \frac{y_t - \mu_Y}{\sigma_Y} \tag{9}$$

After this, the tensors are collated into a batch and potentially moved to the GPU.

Prediction pipeline The model predicts the mean and the covariance for each time steps for the standardized variables. This needs to be converted back to be scale of the variable to be used for imputation. This operation should scale the whole distributions and not only the mean of the prediction. The standardized prediction $\hat{y}_t^z$ is distributed $p(\hat{y}_t^z) = \mathcal{N}\left(\hat{y_t}^z; \mu_{\hat{y}_t}^z, \Sigma_{\hat{y}_t}^z\right)$, the prediction in the original scale $\hat{y}_t$ is distributed $p(\hat{y}_t) = \mathcal{N}\left(\hat{y}_t; \mu_{\hat{y}_t}, \Sigma_{\hat{y}_t}\right)$. From the inverse of Equation 9:

$$\hat{y}_t = \Sigma_Y \hat{y}_t^z + \mu_Y$$

where $\Sigma_Y = \text{diag}(\sigma_Y)$. Then, using the properties of the linear projections of Gaussian distributions:

$$p(\hat{y}_t) = \mathcal{N}\left(\hat{y}_t; \Sigma_Y \mu_{\hat{y}_t}^z + \mu_Y, \Sigma_Y \Sigma_{\hat{y}_t}^z \Sigma_Y^\top\right)$$

2.5 Model training

The available data is split between training and validation set, the first 80% of the data points are used for training, the remaining 20% for validation. The split is not random, so the validation set does not contain periods of time close to the ones used for training.

The KF is initialized with 9 dimensions for the observations (one for each variable). For the state 18 dimensions, the first 9 dimensions are the state level while the other 9 are for the state slope of the local trend model. The control has 14 dimensions, with 7 for the control at time t and the other 7 for the control at time $t - 1$.

The model is trained using gradient descend, by minimizing the loss function. It employs the ADAM optimizer [28]. The learning was manually stopped when the training loss started being constant or the validation loss started to increase.

Several versions of the KF are trained using data with different patterns in the gaps. Figure 3 shows the different combinations. Each model version is named in a way to reflect to training conditions and it uses this pattern: *KF-⟨var missing⟩-⟨n var missing⟩-⟨range gap lengths⟩[-⟨modifier⟩].*

Generic model The first model to be trained is a generic model (**KF-Gen-Sin-6_336**), where each data block had a gap in one variable with the length sampled from a uniform distribution between 6 (3 hours) and 336 (1 week). For each gap, only one variable was missing, which was sampled with equal probability from the list of all variables. The shift was sampled from a normal distribution with mean 0 and standard deviation 50. For each block of data in the original data frame, 10 different artificial gaps were created, resulting in a total of 4080 data blocks used for training and 520 for validation. The length of the block of data was 446, so that at least 50 observations were available to the model before and after the gap. The batch size was 20. The model was trained for 3 epochs with a learning rate of 1×10^{-3}.

Variable fine-tuning The generic model was fine tuned for each variable (**KF-⟨var⟩-Sin-6_336**), resulting in 9 different models. The training settings were the same for the generic model, expect that the gaps were only in one variable and then number of repetitions for each was 5 (training 2040 blocks, validation 260 blocks). Each variable was fine-tuned with a different combination of epochs and learning rate (lr):

- `TA` 3 epochs with lr 1×10^{-3} and 1 epoch with lr 1×10^{-5}
- `SW_IN` 7 epochs with lr 1×10^{-3}
- `LW_IN` 3 epochs with lr 1×10^{-3}, `VPD` 6 epochs with lr 1×10^{-3}
- `VPD` 5 epochs with lr 1×10^{-3}
- `WS` 3 epochs with lr 1×10^{-3}
- `PA` 3 epochs with lr 1×10^{-3} and 1 epoch with lr 1×10^{-5}
- `P` no additional training
- `TS` 6 epochs with lr 1×10^{-3}
- `SWC` 8 epochs with lr 1×10^{-3} and 1 epoch with lr 1×10^{-5}

Short gaps The numerical stability issue limits the gap length to 30 observations (15 hours) if all variables are missing. Therefore, an additional set of models was trained with for short gaps up to 15 hours, in order to have multiple variables missing. A version of the model was trained only with gaps in all variables (**KF-Gen-All-6_30**). The length of the gap ranged from 6 to 30 (3 to 15 hours). The training set contained 7000 unique data blocks and 1760 for validation. The training started from the generic model and lasted for 3 epochs, with a learning rate of 3×10^{-4}. The generic model was also fine-tuned for short gaps (**KF-Gen-Sin-6_30**). The training was for 3 epochs with a learning rate of 3×10^{-4}.
Another version of the model was trained with gaps in any number of variable (**KF-Gen-Multi-6_30**). The number of variables missing was drawn from a uniform distribution from 1 to n, and the variables missing sampled with equal probability. The total gap length ranged from 6 to 30 (3 to 15 hours). For each original data block, 20 different artificial gaps were generated for a total of 28000 blocks in the training set and 7020 in validation. The model was trained starting from *KF-Gen-Sin-6_336* for 3 epochs with a learning rate of 5×10^{-4} and then 1 epoch with a learning rate of 1×10^{-5}.

Additional version Two more model have been trained, with the difference being in the model characteristics instead of the data.
A model has been initialized with random parameters (**KF-Gen-Multi-6_30-Rand**), drawn from a uniform distribution between 0 and 1. The data used for training was the same of *KF-Gen-Multi-6_30*. The model was trained for 3 epochs with a lr of 1×10^{-3} and 3 epochs with a lr of 1×10^{-4}.
For the last version of the model the use of the control variables was disabled (**KF-Gen-Sin-6_336-No_Contr**). The data was the same of *KF-Gen-Sin-6_336* and the training was from scratch for 3 epochs with a learning rate of 1×10^{-3}.

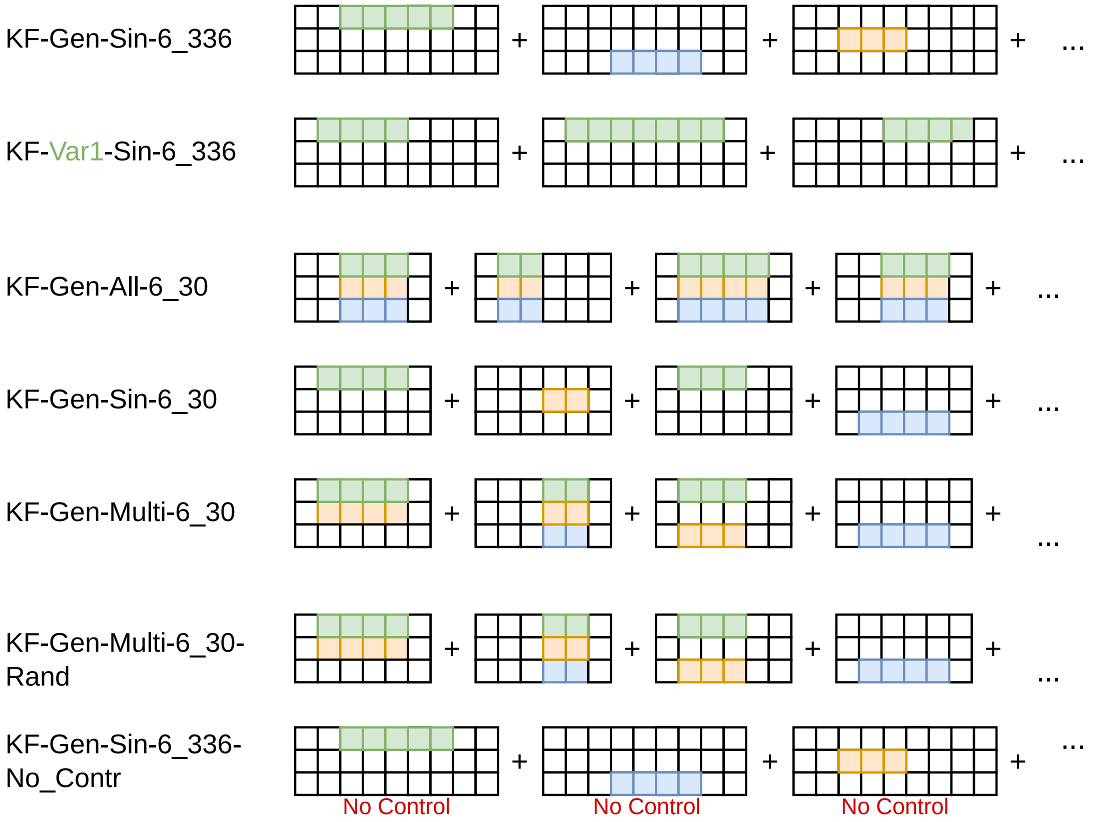


Figure 3: Schematic representation of gap pattern for training scenarios. Each rectangle is a data block used for training, where each row is a different variable and each column a different time. The highlighted areas represent artificial gaps. In the figure, only three variables and a small number of data points are shown. The name of the training scenarios follows this pattern: *KF-⟨var missing⟩-⟨n var missing⟩-⟨range gap lengths⟩[-⟨modifier⟩].*

2.6 Other methods

The implementation of the MDS used in the results comparison is from REddyProc [50]. This package has been used because it provides an R interface, that can be easily integrated with Python. Conversely, ONEFlux implements only a C interface, whose integration in Python is significantly more challenging. The MDS algorithm in REddyProc and ONEFlux are fully equivalent. In detail, the function `REddyProc::sEddyProc_sMDSGapFill` was used, with the default settings of using `SW_IN`, `TA` and `VPD` as drivers with a tolerance of 2.5 °C, 50 W/m^s and 5 hPa respectively as described in Reichstein et al. [43]. The data provided to MDS had a context of at least 90 days around the gap, as required by REddyProc.

The imputation using ERA-Interim was performed by using the ERA-I variables available in the FLUXNET 2015 dataset without further correction.

2.7 Code Details and Availability

The code for this project has been developed in Python. The main libraries used are: PyTorch [40] for the model, FastAI [24] for model training and data preparation, Altair [48, 45] for plotting and Pandas and Polars for data analysis. The source code and the trained model are available at `https://github.com/mone27/meteo_imp`, and the documentation of the library at `https://mone27.github.io/meteo_imp/lib`.

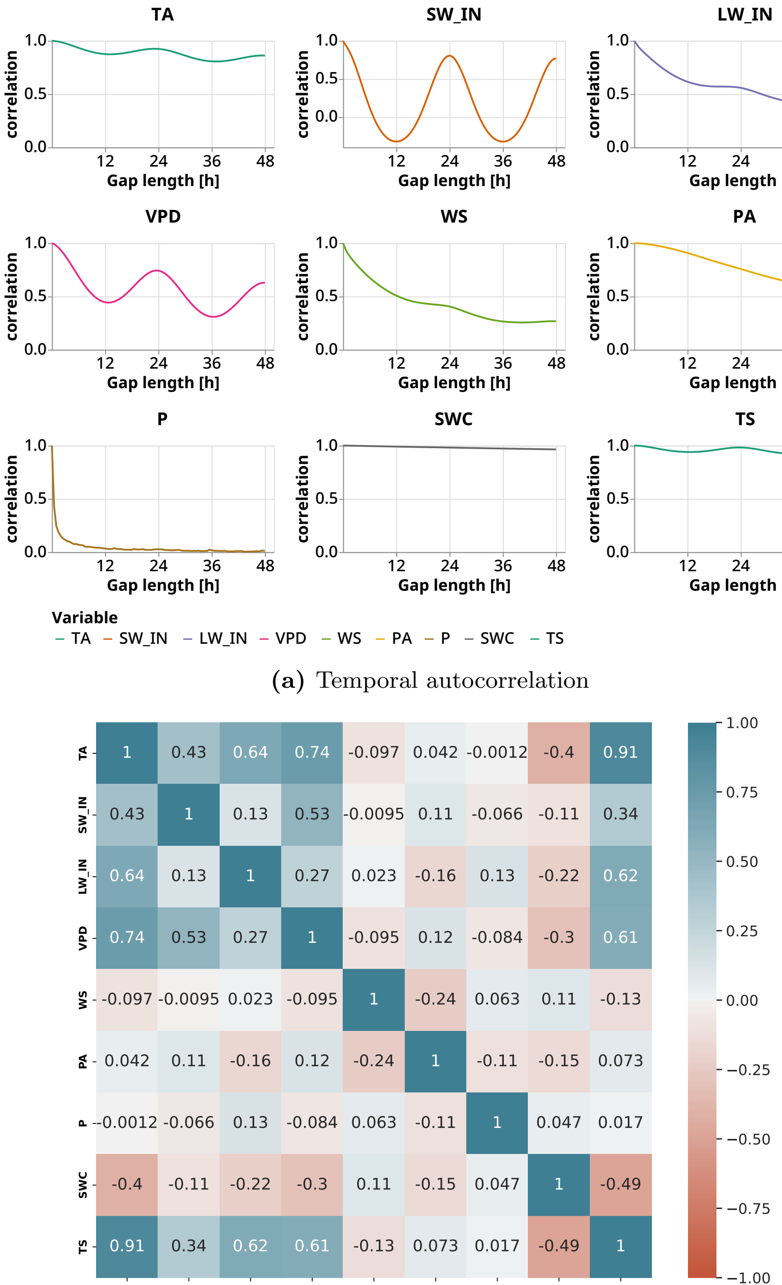
3 Results

3.1 Correlation characteristics of meteorological variables

The analysis of the patterns in the variable temporal autocorrelation and inter-variable correlation supports the interpretation of the results on the performance of imputation methods, as it highlights which mechanisms are available to the models to impute each variable.

The temporal autocorrelation, for a lag up to 48 hours, is shown in Figure 4a. Overall, the meteorological variables have a high temporal autocorrelation, that decreases with the lag. The only exception is the Precipitation (`P`), which has a very low temporal autocorrelation. Moreover, several variables (i.e. Air Temperature `TA`, Incoming Shortwave Radiation `SW_IN`, Vapor Pressure Deficit `VPD`, Soil Temperature `TS`) have a daily pattern with the highest temporal autocorrelation for lags that are multiple of 24 hours and the lowest for lags that are multiple of 12 hours. This is particularly evident in `SW_IN`, that has a negative correlation for a lag of 12 hours.

The variable with the highest correlation with other variable is `TA` (Figure 4b), which is correlated with five other variables (correlation coefficient bigger than 0.4). `TS` is highly correlated with the air temperature, thus follows a similar pattern. Four variables: `SW_IN` `LW_IN`, `VPD`, `SWC` have the correlation ranging between 0.4 and 0.6 with at least two other variables, while the remaining three variables: Wind Speed (`WS`), Air Pressure (`PA`) and Precipitation (`P`), have a low correlation with other variables.



(a) Temporal autocorrelation

(b) Inter-variable correlation

Figure 4: Temporal autocorrelation (a) and inter-variable correlation (b) of meteorological variables. Abbreviations: Air Temperature `TA`, Incoming Shortwave Radiation `SW_IN`, Incoming Longwave Radiation `LW_IN`, Vapor Pressure Deficit `VPD`, Wind Speed `WS`, Air Pressure `PA`, Precipitation `P`, Soil Temperature `TS`, Soil Water Content `SWC`.

3.2 Comparison to other imputation methods

The Kalman Filter (KF) has an overall better imputation performance than the state-of-the-art imputation methods: ERA-I and MDS. In general, across all tested scenario it exhibits the same pattern: KF is the method with the smallest imputation error, ERA-I is the second-best method while MDS is the approach with the highest imputation error (Figure 5 and Table 3). The only exception is precipitation, for which the three methods are equivalent.

I compared the models by creating artificial gaps in a single variable with four different gap lengths (i.e. 6 hours, 12 hours, 1 day, 1 week). For each combination of variable and gap length, 500 artificial gaps were generated and imputed using the three methods. The performance was measured using the RMSE (Figure 5 and Table 3) and the standardized RMSE (appendix Figure A.1 and Table A.1). In addition, the imputation performance was compared visually using three different example time series for all the combinations of each variable and three gap lengths: 6 hours, 12 hours and 1 week (Figures 6, 7 and appendix Figures A.2 A.3, A.4, A.5). For each variable, the fine-tuned KF model has been used (*KF-⟨var⟩-Sin-6_336*).

The average error reduction across all variables and gap lengths is 33% compared to ERA-I and 57% compared to MDS, if precipitation is excluded. The improvement of the KF compared to ERA-I (i.e. the best performing state-of-the-art model) strongly depends on the variable analyzed and the gap length, ranging from an average of 54% for `TA` to 5% for `LW_IN`.

Furthermore, by analyzing multiple artificial gaps for the same conditions (i.e. missing variable and gap length) the maximum value and the standard deviation of the gaps' RMSE is computed. For virtually every scenario, the KF is the method with the smallest maximum and standard deviation of the RMSE (Figure 5 and Table 3).

The imputation performance between different variable can be compared by analyzing the standardized RMSE, which measure the imputation error relative to the variable's standard deviation. There is a wide range in imputation performance, with the standardized RMSE of the worst variable being one order of magnitude bigger of the one of the best variable. The variables with the best imputation quality are: Air Pressure (`PA`, $\text{RMSE}_{\text{stand}} = 0.06$), Air Temperature (`TA`, $\text{RMSE}_{\text{stand}} = 0.08$) and Soil Water Content (`SWC`, $\text{RMSE}_{\text{stand}} = 0.09$). Soil Temperature (`TS` , $\text{RMSE}_{\text{stand}} = 0.14$), Vapor Pressure Deficit (`VPD`, $\text{RMSE}_{\text{stand}} = 0.17$), Incoming Shortwave Radiation (`SW_IN`, $\text{RMSE}_{\text{stand}} = 0.25$) and Incoming Longwave Radiation (`LW_IN`, $\text{RMSE}_{\text{stand}} = 0.33$) have an intermediate imputation quality. The variables with the highest imputation error are: Wind Speed (`WS`, $\text{RMSE}_{\text{stand}} = 0.47$) and Precipitation (`P`, $\text{RMSE}_{\text{stand}} = 0.67$).

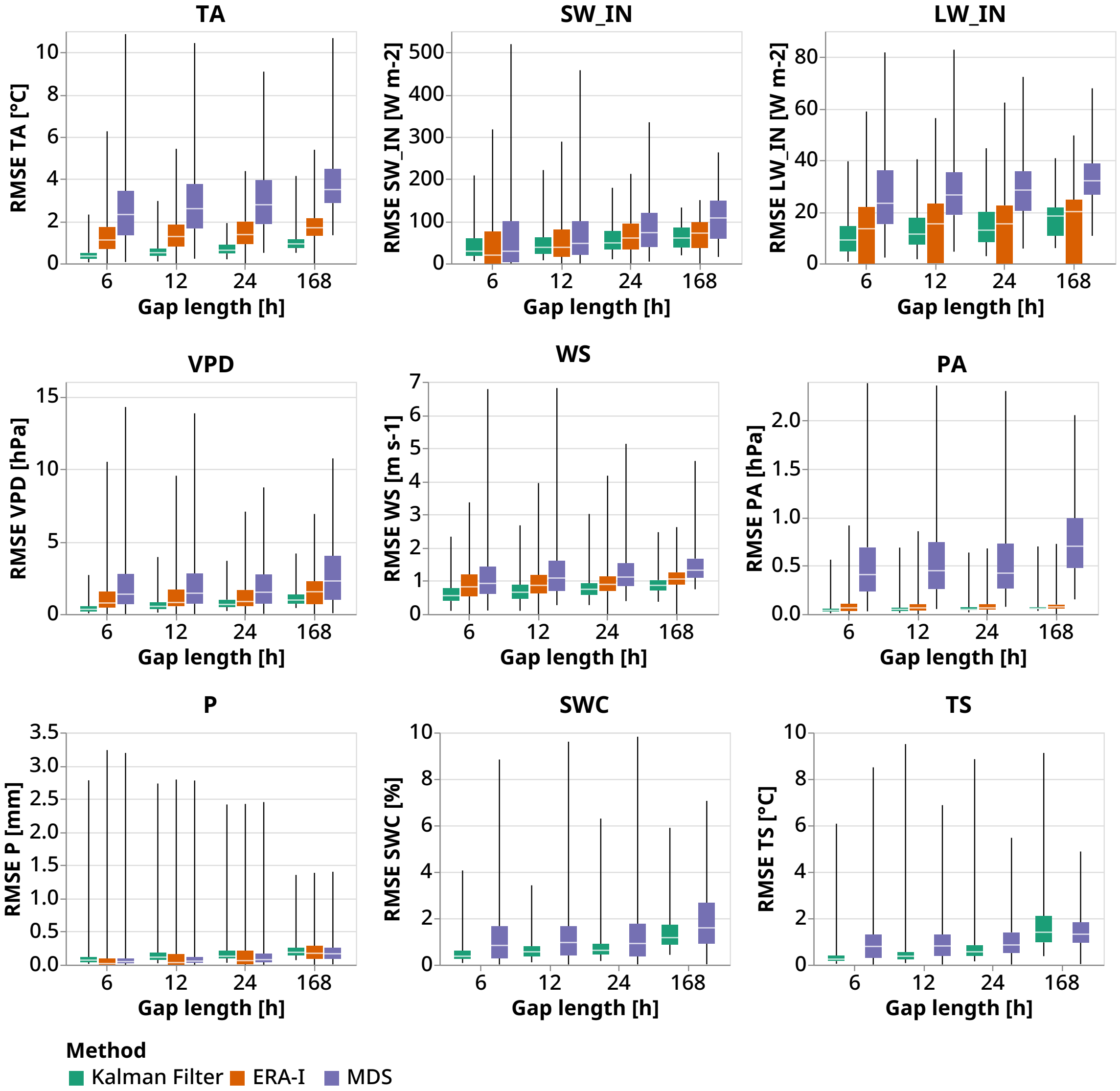


Figure 5: Imputation performance of the Kalman Filter (in green) in comparison to the state-of-the-art methods: ERA-Interim (ERA-I, in orange) and Marginal Distribution Sampling (MDS, in purple). The performance was assessed by calculating for each method the *Root Mean Square Error* (RMSE) for an artificial gap, with a single variable missing. For each combination of variable and gap length a sample of 500 random gaps was used (total of 18000 artificial gaps). The KF model was fine-tuned to each variable (*KF-⟨var⟩-Sin-6_336*). ERA-I dataset does not contain `TS` and `SWC`, so it cannot be used for their imputation. The extent of the box plot vertical lines represent the maximum and minimum value of a gap RMSE.

Table 3: Imputation performance of the Kalman filter in comparison to the state-of-the-art methods: ERA-Interim (ERA-I) and Marginal Distribution Sampling (MDS), using mean and standard deviation of the *Root Mean Square Error* (RMSE). The best method for each gap length is highlighted in bold. For each combination of gap length and variable, 500 artificial gaps were created.

		Kalman Filter		ERA-I		MDS	
	RMSE	mean	std	mean	std	mean	std
Variable	Gap						
TA [C]	6 h	**0.405**	0.258	1.347	0.998	2.713	1.897
	12 h	**0.607**	0.401	1.472	0.901	2.942	1.748
	1 day (24 h)	**0.741**	0.368	1.530	0.800	3.013	1.611
	1 week (168 h)	**1.021**	0.445	1.754	0.643	3.780	1.315
SW_IN [W/m^2]	6 h	**44.637**	40.465	49.333	66.242	63.537	85.402
	12 h	**48.155**	33.868	54.208	49.769	69.427	68.936
	1 day (24 h)	**56.564**	30.043	65.950	40.931	86.771	59.604
	1 week (168 h)	**61.583**	25.740	70.224	34.883	107.384	53.606
LW_IN [W/m^2]	6 h	**10.902**	7.736	13.805	12.988	26.680	15.022
	12 h	**13.422**	7.735	14.767	12.585	28.085	13.457
	1 day (24 h)	14.594	7.840	**14.093**	12.228	29.614	12.417
	1 week (168 h)	17.063	6.425	**16.366**	11.130	32.955	8.834
VPD [hPa]	6 h	**0.428**	0.363	1.297	1.547	2.084	2.149
	12 h	**0.661**	0.505	1.265	1.289	2.137	2.096
	1 day (24 h)	**0.828**	0.502	1.248	1.032	1.912	1.605
	1 week (168 h)	**1.126**	0.633	1.662	1.127	2.661	1.965
WS [m/s]	6 h	**0.617**	0.317	0.912	0.508	1.136	0.783
	12 h	**0.715**	0.351	0.957	0.524	1.261	0.797
	1 day (24 h)	**0.802**	0.343	0.949	0.447	1.276	0.609
	1 week (168 h)	**0.950**	0.363	1.089	0.349	1.495	0.615
PA [hPa]	6 h	**0.045**	0.034	0.075	0.062	0.531	0.441
	12 h	**0.053**	0.042	0.077	0.058	0.564	0.427
	1 day (24 h)	**0.059**	0.039	0.079	0.051	0.557	0.404
	1 week (168 h)	**0.066**	0.048	0.084	0.054	0.773	0.384
P [mm]	6 h	0.134	0.274	**0.113**	0.316	0.118	0.306
	12 h	0.179	0.295	0.139	0.297	**0.130**	0.281
	1 day (24 h)	0.206	0.254	0.166	0.288	**0.159**	0.265
	1 week (168 h)	0.240	0.174	0.223	0.202	**0.215**	0.197
SWC [%]	6 h	**0.508**	0.487	-	-	1.314	1.557
	12 h	**0.665**	0.472	-	-	1.278	1.323
	1 day (24 h)	**0.779**	0.641	-	-	1.356	1.472
	1 week (168 h)	**1.494**	0.948	-	-	1.948	1.488
TS [C]	6 h	**0.341**	0.432	-	-	0.954	0.889
	12 h	**0.534**	0.784	-	-	1.003	0.877
	1 day (24 h)	**0.787**	0.852	-	-	1.078	0.857
	1 week (168 h)	1.660	1.078	-	-	**1.440**	0.764

Air Temperature Air temperature is the variable where there is the biggest improvement in performance when using KF. It outperforms ERA-I for all gap lengths, with an average reduction in the RMSE of 54%, while compared to MDS the reduction in RMSE is of 77%. Comparing KF with ERA-I, the relative improvement in RMSE decrease with the gap length (69 % for 6 hours gaps, 41% for 1 week long gaps), while the average difference in RMSE increases with the gap length (0.29 °C for 6 hours gaps, 0.34 °C for 1 week long gaps). The visual inspection of the time series indicate an overall very good reconstruction of the missing data by the KF.

Incoming Shortwave Radiation The KF is the method with the smaller average RMSE, but the relative improvement is small, only 12% compared to ERA-I. The highest error is at night, where `SW_IN` is by definition always 0, but the KF often predicts sudden changes and negative values with errors in the order of 50 W/m^2. However, during the day, which is the most important condition in `SW_IN` imputation, the KF is better than other methods. This is also confirmed by visual inspections of the time series.

Incoming Longwave Radiation The imputation performance of the KF is comparable with the one of ERA-I. However, KF is the best method for short gaps (20 % improvement for 6 hours long gaps). For instance, this can be visualized in the 12 hours gap in Figure A.4. The imputation performance of MDS is poor, especially for long gaps.

Vapor Pressure Deficit KF is the best model for all gap lengths. The relative performance is higher for short gaps (66 % compared to ERA-I for 6 hours long gaps) and progressively smaller for longer gaps (32 % compared to ERA-I for 1 week long gaps). The analysis of the time series suggest that the KF is overall good at reconstructing the higher frequency changes of `VPD`, but in some scenarios KF incorrectly predicts short term variation (e.g. 1 week long gap in Figure A.2).

Wind Speed The KF is the best imputation method for all gap lengths, with an average error reduction of 21% compared to ERA-I. The `WS` is the variable with the second highest standardized RMSE (appendix Figure A.1), which indicates that the RMSE is high compared to the `WS` standard deviation. The visual inspection of the time series (Figures 6 and A.2), indicates that the `WS` has often a high variability on short time scales, which is not captured neither by ERA-I nor by the KF.

Air Pressure The imputation error of `PA` is low for KF and ERA-I. The KF outperforms ERA-I with an improvement ranging from 36% for 6 hours long gaps to 20% for 1 week long gaps. The visual analysis of the time series suggests that the KF slightly overestimates the short term variability for `PA`. MDS is significantly worse, with the imputation error one order of magnitude bigger than the one of the other methods.

Precipitation The Precipitation is a variable where no method performs well, with models in some case predicting precipitation event that do not exist and in other missing the real precipitation. The RMSE of the three methods is comparable. However, the RMSE is not an accurate metric for `P` imputation, due to the very high number of data points with no precipitation (over 90% in Hainich). For reference, the RMSE of a null model (i.e. always predicts 0) is 0.28 mm, which is comparable with the errors of all imputation methods. The visual analysis of the time series shows that ERA-I predictions are the only one that are physically realistic, even though the precipitation amount is often incorrect, while the KF often predicts negative values for P, which is physically impossible, and MDS in a tested scenario (appendix Figure A.3) predicts an unlikely constant low amount of P for a long period.

Soil Water Content The KF is the best imputation method, for short gaps there is an error reduction of 61% compared to MDS, while for long gaps (1 week) the absolute error of the KF roughly doubles and the improvement in performance is limited to 23%. SWC is a variable that is not available in ERA-I, so KF and MDS are the only available methods and the KF does not have access to a control variable for SWC. The analysis of the time series shows that the mean of KF prediction is overall accurate, and notably it manages also to correctly predict sudden changes in SWC (Figre 7). However, the KF constantly predicts small variations in the soil water content, which are not reflected in the observations.

Soil Temperature The KF is the best imputation method for short gaps (less than 24 hours), but the MDS is better for 1 week long gaps (Figure 5). For short gaps there is a big difference in the methods' error (up to 60%), but is reduced for long gaps (KF is 15% worse than MDS for 1 week long gaps). TS is not available in the ERA-I dataset. In two of the long time series analyzed the TS is almost constant (1 week long gaps in Figure 7 and A.5), but the KF incorrectly predicts important variations, while the MDS is overall constant. In another scenario (appendix Figure A.3), where there is a diurnal pattern in TS, the KF predictions have an overall correct shape, even though there is roughly 1 °C error.

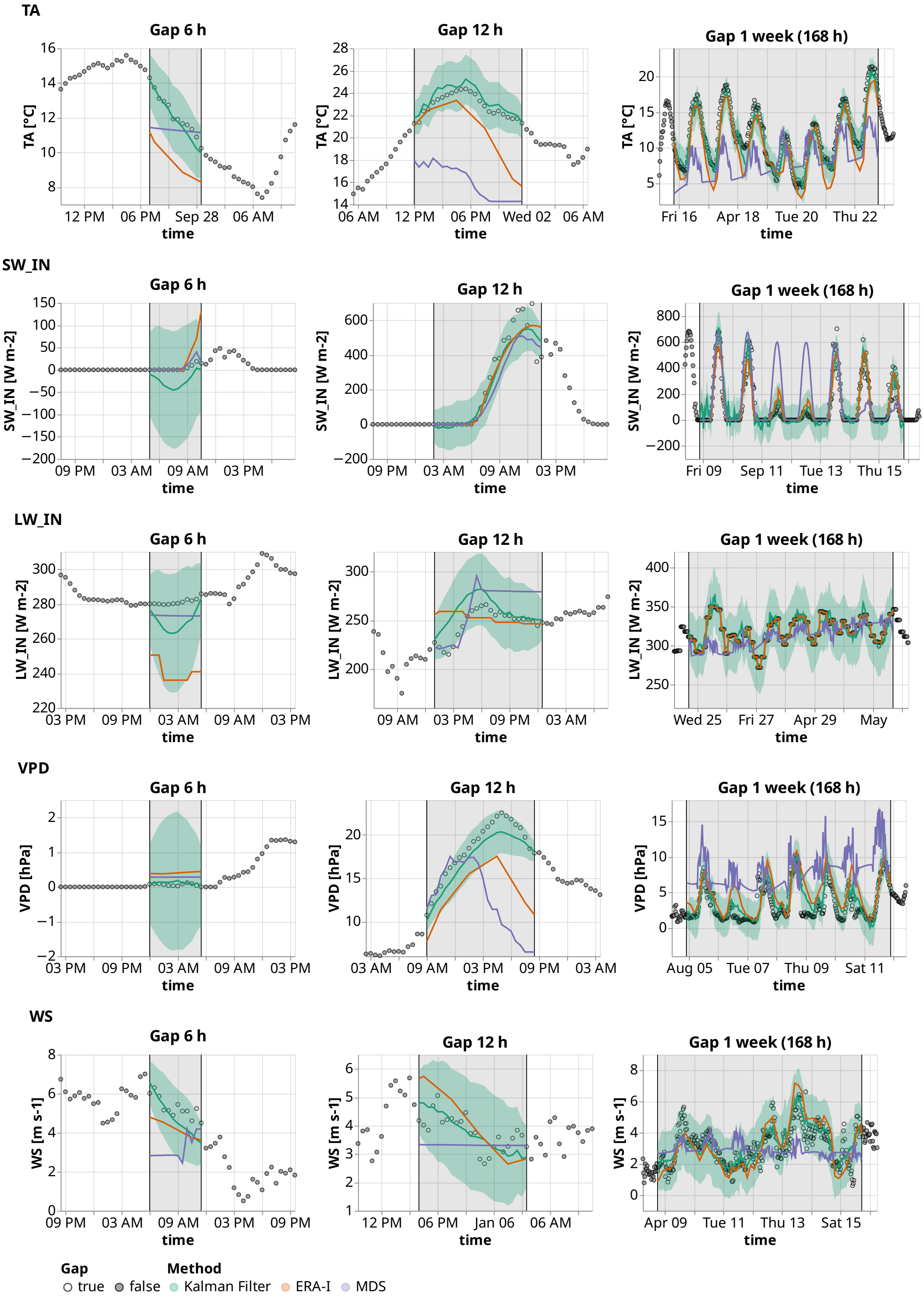


Figure 6: Time series to visualize the imputation of TA, SW_IN, LW_IN, VPD, WS using different methods: Kalman Filter (KF), ERA-Interim (ERA-I) and Marginal Distribution Sampling (MDS). For each variable, three random artificial gap (length 6 hours, 12 hours, 1 week) are imputed using the three methods: Kalman Filter (green), ERA-I (orange), MDS (purple). For the KF the shaded area shows the uncertainty of the prediction ±2σ (standard deviation). The grey shaded area and the vertical black lines delimit the artificial gaps, where the observations are not available to the model but are used to assess the imputation performance. The ERA-I prediction is the control variable of the KF. The KF model has been fine-tuned to each variable (*KF-⟨var⟩-Sin-6_336*).

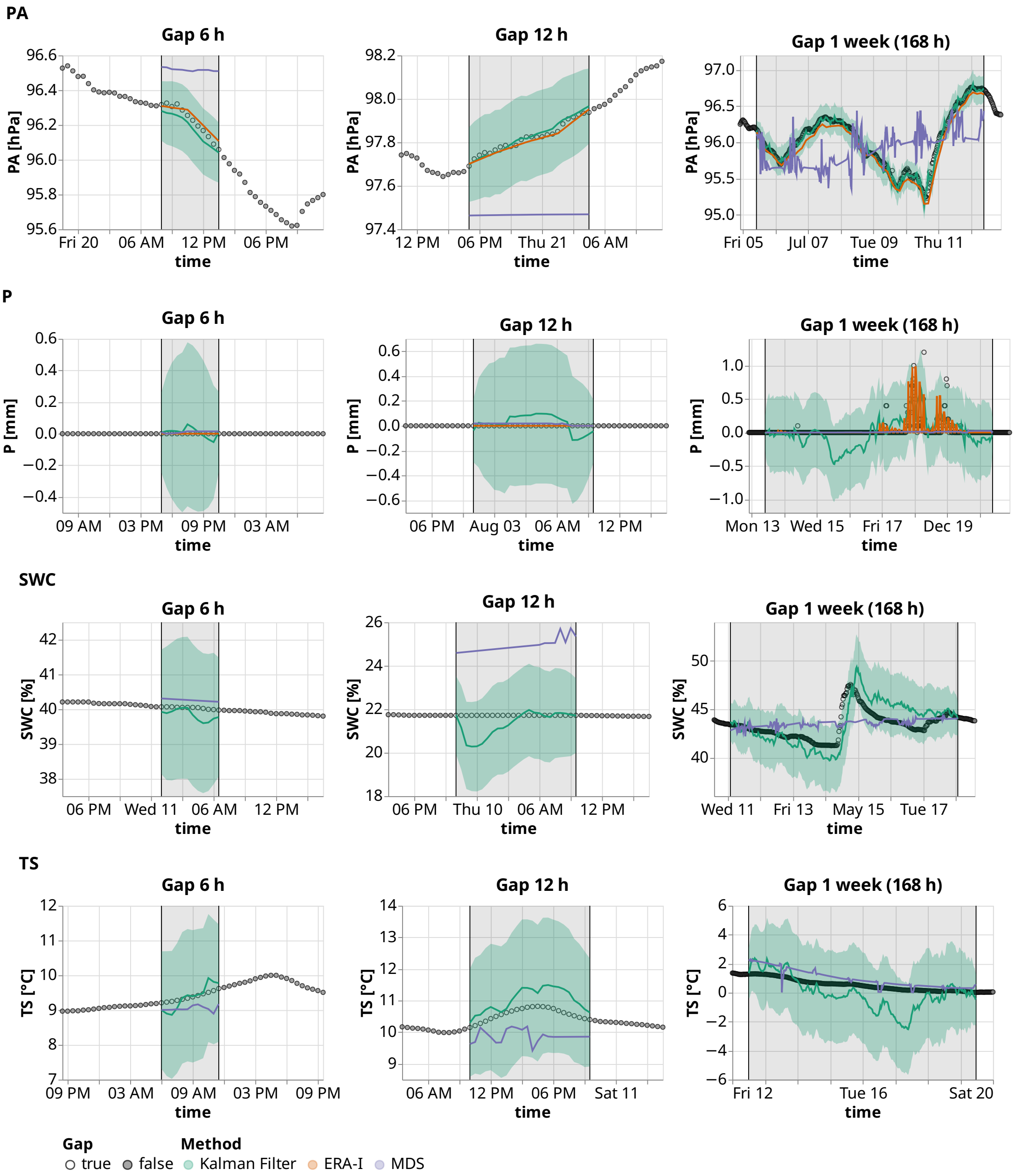


Figure 7: Time series to visualize the imputation of `PA`, `P`, `SWC`, `TS` using different methods: Kalman Filter (KF), ERA-Interim (ERA-I) and Marginal Distribution Sampling (MDS). For each variable, three random artificial gap (length 6 hours, 12 hours, 1 week) are imputed using the three methods: Kalman Filter (green), ERA-I (orange), MDS (purple). For the KF the shaded area shows the uncertainty of the prediction $\pm 2\sigma$ (standard deviation). The grey shaded area and the vertical black lines delimit the artificial gaps, where the observations are not available to the model but are used to assess the imputation performance. The ERA-I prediction is the control variable of the KF. The KF model has been fine-tuned to each variable (KF-$\langle \mathit{var}\rangle$-$\mathit{Sin}$-$\mathit{6\_336}$).

3.3 Aspects affect Kalman Filter performance

Gap Length The effect of the gap length on the KF imputation performance was analyzed by measuring the RMSE on gaps of a single variable for lengths ranging from 1 hour to 1 week (Figure 8 and appendix Table A.2). For the majority of the variables the error increases with the gap length only up to 24 hours and then is almost constant. For three variables (P, SWC and TS) the imputation error keeps increasing after 24 hours, but the rate decreases with the gap length. The highest rate of change of the imputation error is between 1 hour long gaps and 12 hours long gaps. For all variables, there is a significant difference in the error of very short gaps (1 hour) and long gaps (1 week), on average the error for gaps 1 hour long is 31% of the one for 1 week long gaps. The variability in the RMSE between gaps (i.e. std in appendix Table A.2) increases with the gap length of the majority of the variables (i.e. TA, VPD, PA, SWC, TS). For the other variables (i.e. SW_IN, LW_IN, WS, P), there is a decrease in RMSE variability with increases in gap length.

Control variables The importance of the control variables has been assessed by comparing the imputation error of a KF model that used the control variables (*KF-Gen-Sin-6_336*) with a KF model that did not have access to the control variables (*KF-Gen-Sin-6_336-No_Contr*). Both models were not fine-tuned for each variable.

In general, the use of control variables improves the imputation performance for all variables for all gap lengths. The exceptions are P and TS, where the two models are equivalent, and for short gaps (6 hours) in SWC and WS, where the model with the control performs worse than the one without control (Figure 9). For all variables, the longer the gap, the bigger the performance improvement of the model with the control variables (appendix Table A.3). The variable that benefits the most from the control is PA, where for gaps 1 week long, the model without the control has an error almost 6 times bigger than the one with the control.

Gaps in multiple variables The importance of the inter-variable correlation for the KF predictions was assessed by comparing the imputation for a gap with only one variable missing and then for the same gap with all variables missing. All the gaps were imputed using the same model (*KF-Gen-Multi-6_30*). The gap length was limited to 15 hours due to numerical stability issues.

The presence of other variables in the gap is overall improving the model predictions (Figure 10 and appendix Table A.4), for some variables there is a significant error reduction (e.g. around 40 % for TA) but for others the improvement is minimal (e.g. less than 2% for WS). Across the different variables, there is an increase in the absolute values of the difference in RMSE with an increase in gap length.

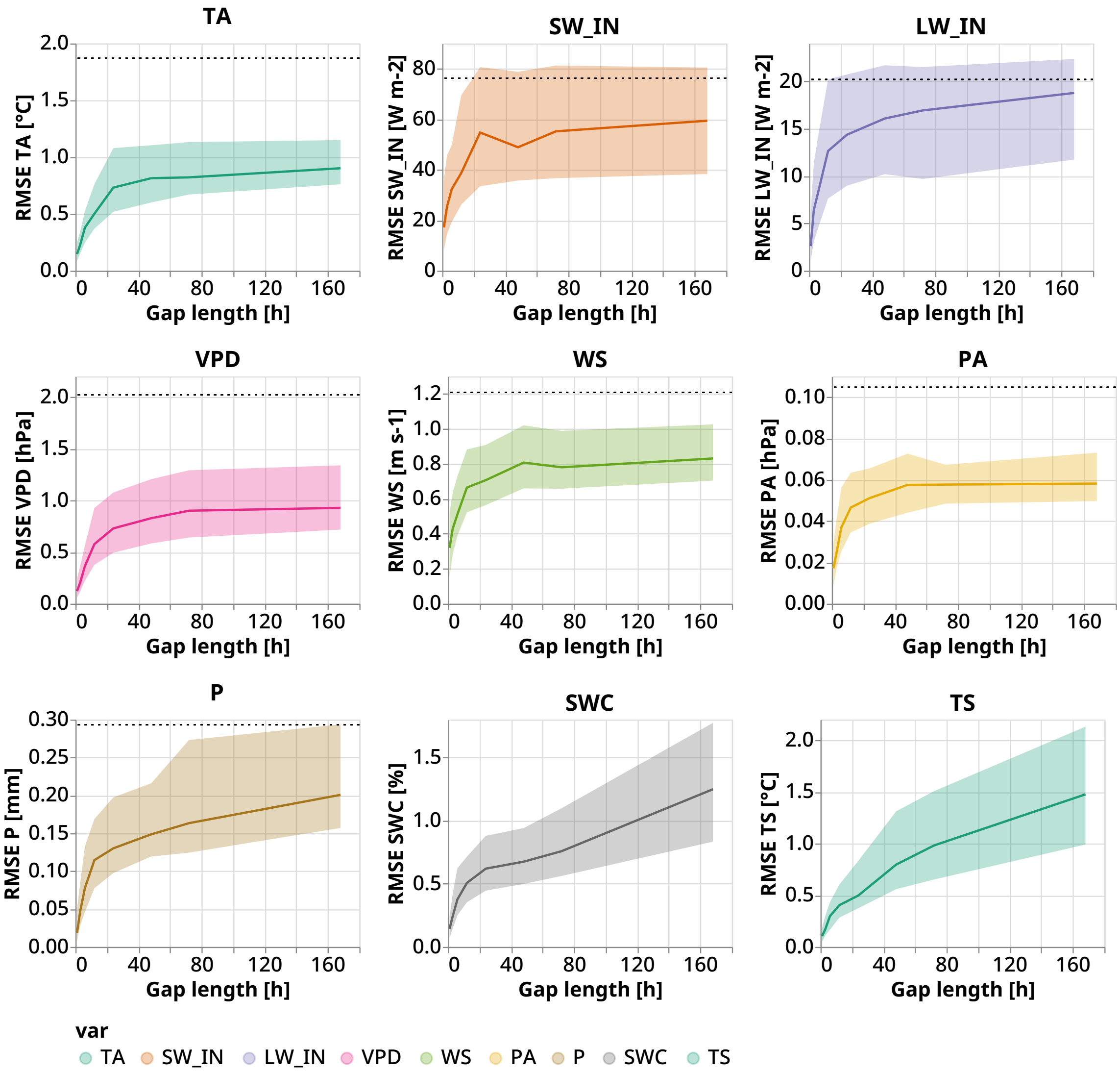


Figure 8: Effect of gap length on the KF performance. The solid line shows the median RMSE, while the shaded area is delimited by the first and third quartile. The dotted black line is the mean ERA-I error for the entire dataset (ERA-I data is not available for `SWC` and `TS`). Seven different gaps lengths were tested (1 hour, 3 hours, 6 hours, 12 hours, 1 day, 2 days, 3 days, 1 week), for each of them 500 artificial gaps were generated for every variable (total 31500 gaps). For each variable, the fine-tuned KF model has been used (*KF-⟨var⟩-Sin-6_336*).

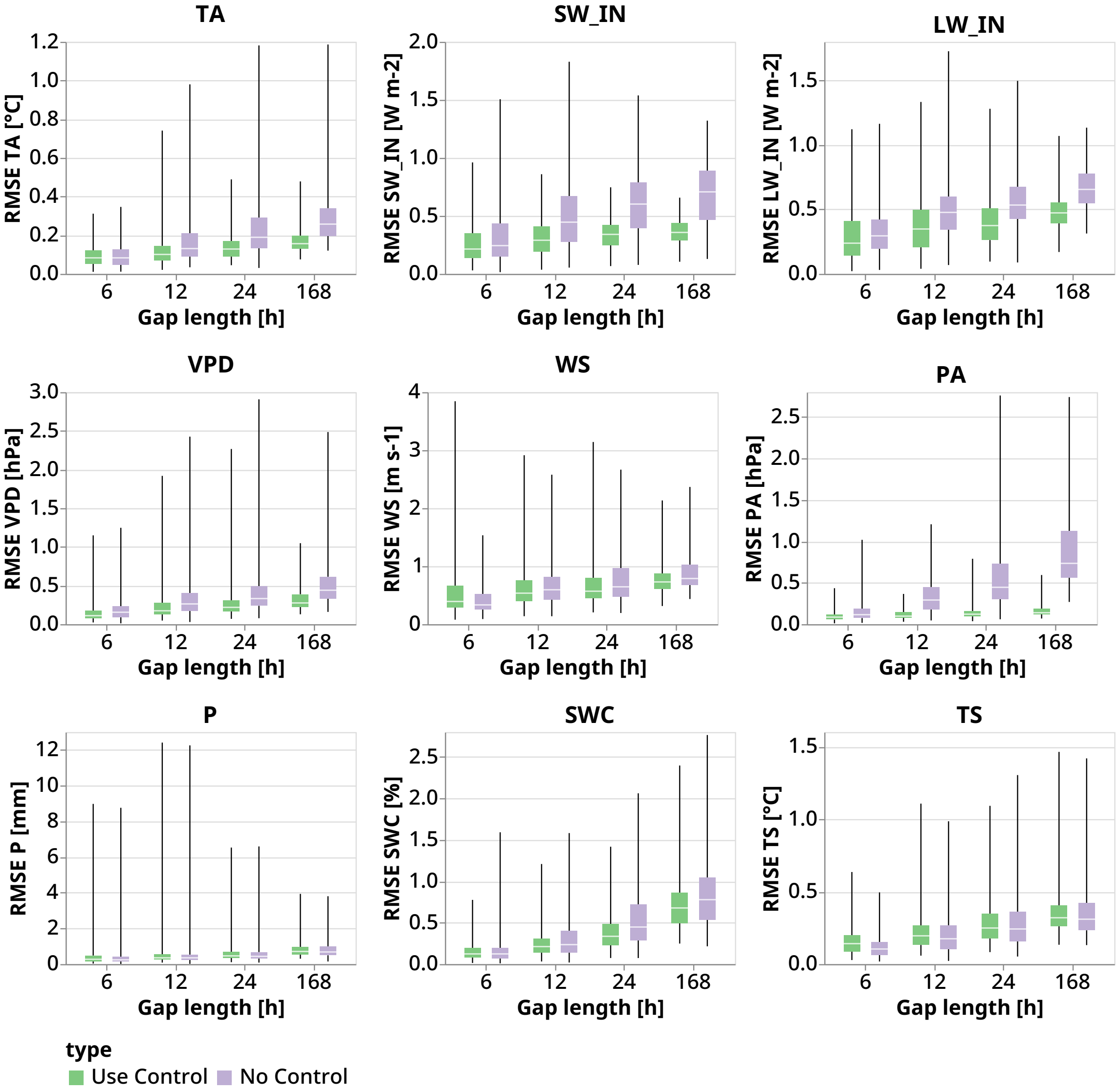


Figure 9: Comparison of imputation performance between KF with control variables (in green, *KF-Gen-Sin-6_336*) and KF without control variables (in purple, *KF-Gen-Sin-6_336-No_Contr*). For each combination of variable and gap length, 500 artificial gaps were created. The extent of the box plot vertical lines represent the maximum and minimum value of a gap RMSE.

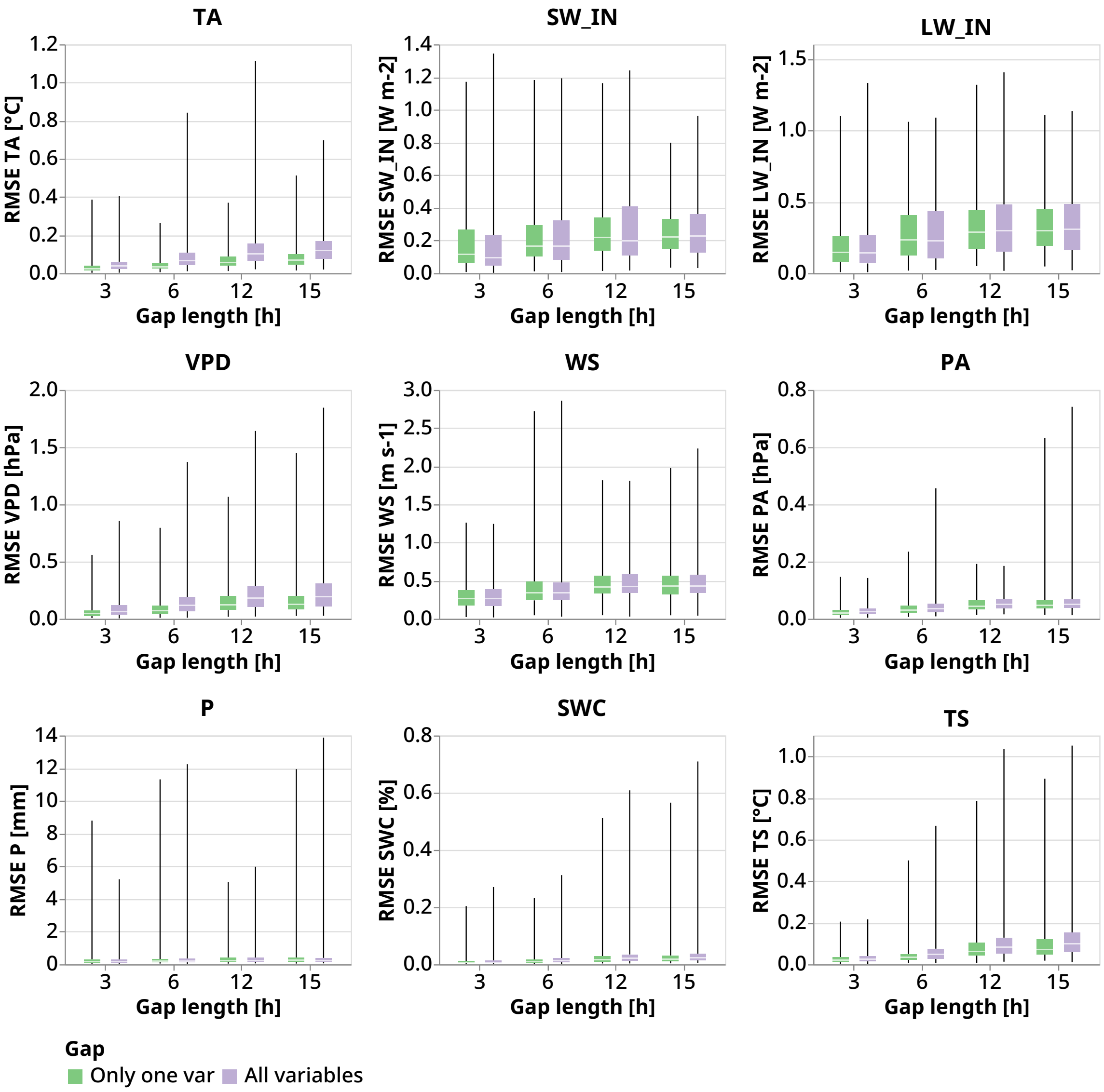


Figure 10: Comparison of imputation performance between a gap in only the variable of interest (green) and the same gap with all other variables missing (purple). The model used for imputation is always *KF-Gen-Multi-6_336*. For each combination of variable and gap length, 500 artificial gaps were created. The extent of the box plot vertical lines represent the maximum and minimum value of a gap RMSE.

3.4 Kalman Filter training

Variable fine-tuning The performance of the KF is improved if the model is fine-tuned on gaps only for one variable (e.g. only for TA). For each variable a different model was used (*KF-(var)-Sin-6_336*) and the performance was compared to a generic model that was trained with gaps in any variable (*KF-Gen-Sin-6_336*).

The fine-tuning is reducing the error for all the variables (Figure 11 and appendix Table A.5). The entity of the error reduction changes depending on the variable, ranging from 75% for SWC to 12% for LW_IN. The precipitation is the only variable that was not fine-tuned, as additional training did not improve the performance. For the majority of variables, the variability of the RMSE between gaps of the fine-tuned model and the maximum RMSE is smaller for the fine-tuned model.

Training limitations During the training of different KF versions, I experienced limitations in the ability of the KF to learn the optimal parameters. For instance, if the KF was initialized with random parameters, it did not achieve the same performance as a KF initialized with the local trend model. The training conditions were the same, so the model that started with random parameters should have been able to learn the same parameters, but this was not the case.

The difficulties to train the KF model are shown in Figure 12. Three models were compared: a model trained with gaps in multiple variables (*KF-Gen-Multi-6_336*), a model trained with gaps of only one variable (*KF-Gen-Sin-6_336*) and a model trained with gaps in multiple variables but initialized with random parameters (*KF-Gen-Multi-6_336-Rand*). All models were trained until there was no improvement in the validation loss. The three models were used to impute a gap with only one variable was missing. The expectation is that they should have comparable performance, but this is not the case (Figure 12 and appendix Table A.6). The training conditions between *KF-Gen-Multi-6_336* and *KF-Gen-Multi-6_336-Rand* were exactly the same, so they should be able to achieve the same performance. The other model, *KF-Gen-Sin-6_336*, was trained on the same conditions that were used for testing, so it should not perform worse than *KF-Multi-Sin-6_336.* The best model is always *KF-Gen-Multi-6_336*, while depending on variables, the second-best model is either *KF-Gen-Sin-6_336* or *KF-Gen-Multi-6_336-Rand.*

Numerical Stability The numerical instability of the current KF implementation limits the gap length to 15 hours long, if all the observations are missing. When only one variable is missing in a gap, I successfully tested models with a gap length of 2 weeks. The numerical instability produces covariance matrices that are not positive definite, which in turns makes the log likelihood computation fail. The numerical stability of the filter is influenced by the values of the parameters, the observed data, the length of the gap and the total number of observations. The Square Root KF effectively mitigates the numerical stability (appendix Figure A.6), but results in a decrease in the runtime performance. When comparing the performance of the filtering pass, the Square Root KF is 37% slower than the Standard KF (appendix Figure A.7), with a setting similar to the ones used in the rest of this work. The higher the number of dimensions or the batch size, the lower the relative performance of the Square Root KF.

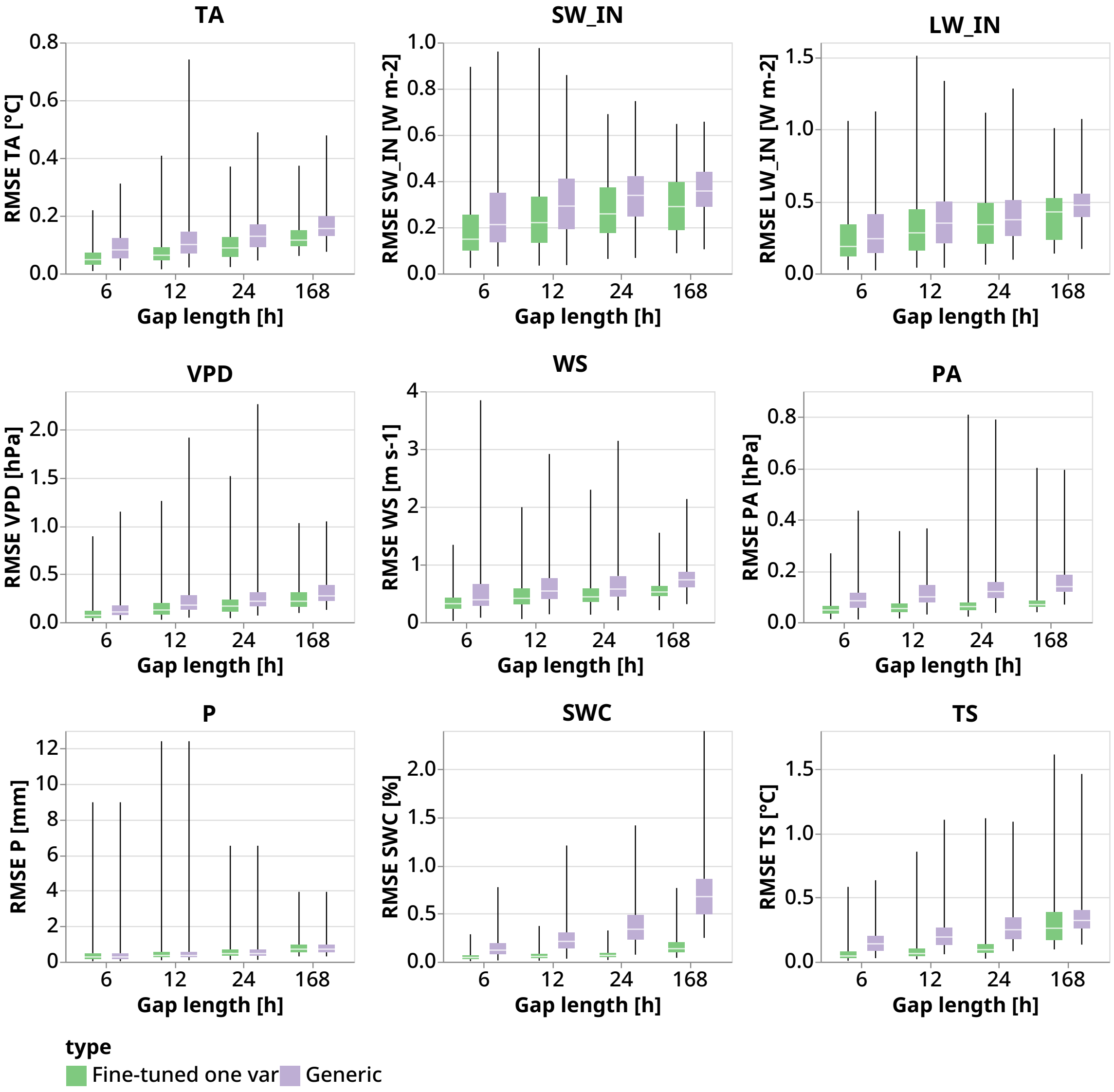


Figure 11: Comparison between KF models fine-tuned to each variable (in green, *KF-⟨var⟩-Sin-6_336*) and generic model trained for gaps in any variable (in purple, *KF-Gen-Sin-6_336*). For each combination of variable and gap length, 500 artificial gaps were created. The extent of the box plot vertical lines represent the maximum and minimum value of a gap RMSE.

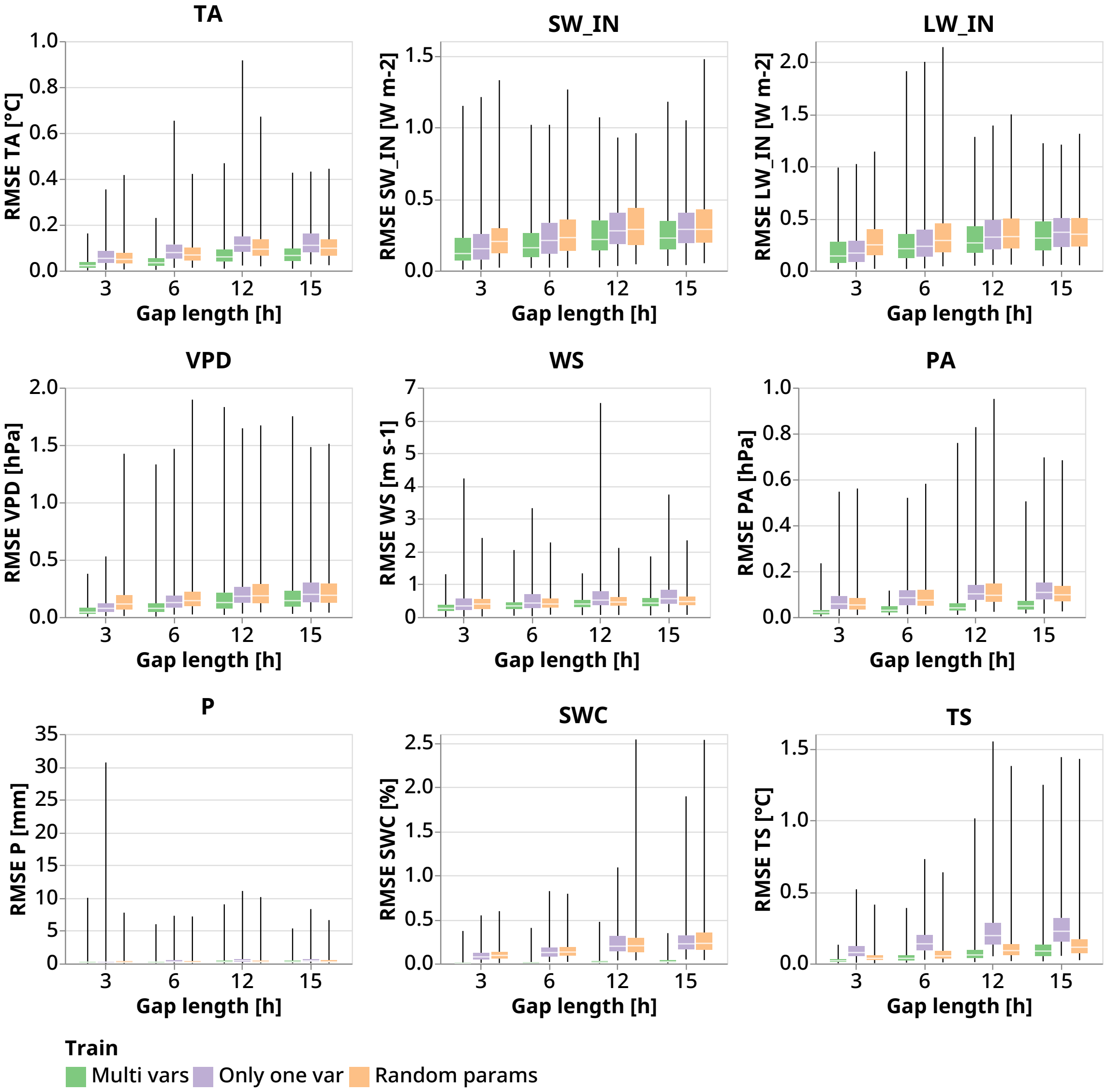


Figure 12: Visualization of KF training difficulties. For a gap in one variable, three models are compared: a model trained with gaps in multiple variables (in green, *KF-Gen-Multi-6_336*), a model trained with gaps of only one variable (in purple, *KF-Gen-Sin-6_336*) and a model trained with gaps in multiple variables but initialized with random parameters (in orange, *KF-Gen-Multi-6_336-Rand*). The models are expected to have comparable performances. For each combination of variable and gap length, 500 artificial gaps were created. The extent of the box plot vertical lines represent the maximum and minimum value of a gap RMSE.

4 Discussion

4.1 Kalman Filter performance

The KF model outperforms the state-of-the-art methods in almost every scenario tested. This result confirms the starting hypothesis that the imputation of EC meteorological time series could be improved by making a more effective use of temporal autocorrelation and by combining the imputation approaches.

The KF models the evolution of the state between consecutive time steps, thus the variable temporal autocorrelation is key for obtaining accurate predictions. The results are consistent with this: the relative performance of the KF is higher for short gaps, and the imputation error of the KF has a strong dependence on the gap length only for gaps shorter than 24 hours. The shorter the gap, the higher the correlation between the observations close to the gap and the missing data and therefore the greater the possibility of an accurate imputation. Moreover, the relative performance of the KF is higher for variables that have a higher temporal correlation (e.g. TA, PA and VPD).

When the control variables are included, the KF performance improves across all variables, confirming the hypothesis that the combination of the imputation approaches can produce better predictions. In line with the expectations, the use of the ERA-I data has an important impact on the imputation of long gaps, though it has limited influence on short gaps. Notably, the use of the control variables improve the predictions also for SWC, even if it is not present in the ERA-I dataset.

The KF model is able to exploit the inter-variable correlation to improve the predictions. Overall, the imputation of variables with the highest correlation benefits the most from the presence of the other variables in the gaps, which matches the expectation. However, this relationship is not true for every variable. For instance, TS has a high correlation with other variables, but I observed only a relatively small increase in performance when all the other variables were available in the gap, compared to gaps where other variables were missing. Further, this analysis was limited to short gaps, but it is reasonable to expect that for longer gaps inter-variable correlation plays a bigger role in the KF predictions.

Shortwave radiation The low performance of the SW_IN at night highlights a limitation of the KF. At its core, the KF considers only changes in the state between consecutive time steps, hence it cannot directly model the daily pattern of SW_IN. The KF can predict this daily change by either using other variables or the control variables. The former is not possible for SW_IN as it is the only variable with a very pronounced daily pattern, therefore the latter approach is the main way the KF can maintain a good imputation performance for longer gaps (see Figure 9). The limited performance of the SW_IN at night can also be explained by the big difference in the rate of change of SW_IN between the day and the night, as the KF uses constant parameters, and thus it does not consider the variability in the SW_IN conditions.

Precipitation The precipitation has a low temporal autocorrelation and a low correlation with other variables. Moreover, it has high temporal and spatial variability [31], especially on short time scales. In fact, ERA-I predictions have a high error for P, which is further worsened by the temporal downscaling. The authors of the paper that introduced ERA-I for EC meteorological imputation [49], considered the timing of the precipitation in ERA-I not accurate enough for a direct comparison of the time series. Therefore, the poor performance of the KF is expected as none of the mechanisms used for imputations (i.e. temporal autocorrelation, inter-variable correlation and control variables) can be effectively applied to gaps in P.

Precipitation has unique characteristics, compared to the other meteorological variables, which make it necessary to employ tailored imputation approaches. For instance, Chivers et al. [12] developed a method specialized in the of imputation `P`, where the prediction is performed in two steps: a first model predicts whether there is going to be precipitation and a second model predicts the amount of precipitation.

Wind Speed The KF does not model accurately the high frequency variation of the wind speed. This is consistent with the limitations of ERA-I [49]. The KF cannot extract the information about the high frequency variation from other observations of the wind sped, KF is able to model higher frequency variations only if the information is present in either the control variable or in other variables. For `WS`, this is not the case, as ERA-I has the same limitation and no other variable has a high correlation with the wind speed. This limitation of ERA-I is also the likely reason behind the increased error in `WS` when using the control for short gap length, where the low variability in ERA-I negatively affects the model prediction. This scenario shows one limitation arising from the simplicity of the KF compared to a Gaussian Process, which should be able to model the high frequency variation in `WS`. Nonetheless, the KF has a better performance than the current imputation methods.

Model Training The KF achieves the best imputation performance only when the model is fine-tuned to each variable. Moreover, I expect that by tuning the model further to more granular conditions the performance would improve. For instance, the rate of change of `TS` is significantly higher in a dry summer day compared to a winter day, when the ground is covered in snow. Therefore, the optimal KF parameters would be different between those conditions. Furthermore, the optimal parameters are likely different between short gaps and long gaps, as in the first case more importance should be given to the temporal autocorrelation, while in the second case inter-variable correlation and control variables should have a bigger weight. Therefore, I expect improvements in the KF performance if the model is trained only on gaps with a similar length. Finally, if the KF is applied to other EC sites, then it will likely require further fine-tuning, as there are different climatic conditions and the error in ERA-I data (i.e. the control variables) is a very different between EC sites [49].

An important outcome of this work is that the KF is able to achieve good performance, but it may be difficult to learn the model parameters. The initialization of the parameters with a local trend model helped to mitigate this issue, however it is probable that the KF parameters trained in this work are still not the optimal ones. For instance, the KF performance for `TS` is relatively poor for long gaps, even though it is a variable with high temporal autocorrelation and correlation with other variables, which suggest that that limiting factor is the ability to learn and not the formulation of the KF. Moreover, for short gaps, the model trained on gaps of multiple variables (i.e. in each gap a different number of variables is missing) outperforms the model trained on gaps with only one variable missing. The numerical stability issue did not allow testing this training condition on longer gaps, but it is likely that a similar pattern would be observed.

4.2 Kalman Filter application

The results of this study indicate that a KF can be employed to improve the imputation of meteorological variables for EC applications. The highest error reduction is for short gaps (less than 1 day), making the KF particular suitable in this scenario. The significant improvement in the imputation of `TA` is likely relevant to reduce error in Land Surface Models. Temperature is a key driver of core ecosystem processes, like photosynthesis or respiration. Those processes have a strong non-linear dependency on temperature [7], hence inaccuracies in the temperature estimation can have substantial impact on the Land Surface Models output.

For medium gaps (1 week) the relative performance of the KF is reduced, but it remains the best imputation approach. The `TS` is the only variables where the KF has a worse performance than state-of-the-art methods. Gaps longer than a week were not tested, but the results suggest that the predictions of the KF is going to get progressively closer to the ERA-I ones with an increase in gap length. However, for the variables not available in ERA-I the performance would probably constantly decrease with gap length.

The KF always provides an interpretable estimate of uncertainty of the predictions. The knowledge of imputation quality of each data point allows data users to make informed decisions regarding if and how to utilize the imputed data depending on each application scenario requirements. This is a significant improvement over the use of a flag system (used in state-of-the-art methods), which is not limited to a fixed set of values.

The current KF implementation has three limitations that would prevent the application in a production scenario: numerical instability, when all variables are missing for more than 15 hours; poor performance for precipitation; physically impossible predictions of `SW_IN` at night. However, I believe that the first and third limitations are relatively straightforward to overcome, as suggested in the following section. Instead, the KF should not be used for precipitation, as it is not suited to its unique characteristics. Moreover, there is no indication that the KF can be improved to effectively predict missing precipitation.

Another aspect that needs to be considered when utilizing a KF is the necessity to fine tune the parameters. A generic KF model, trained on a wide range of conditions, is able to impute gaps, however the best performance is achieved only when the KF parameters are fine-tuned to specific conditions. The need to fine-tune the parameters increases the deployment complexity and potentially the computational cost. There are different approaches to deploying a KF, ranging from using only a generic model, which is the simplest method at the cost of limited performance, to fine-tuning the model to the condition of each individual gap, which would result in the best performance while having the highest computational cost and complexity.

A promising approach is to pre-train models for different conditions (e.g. variable missing, site, time of the year), then for each gap select the KF model trained on the closest conditions. The advantages of this system are the simplicity and the computational efficiency. The only computation overhead arises from the use of variable-specific models in gaps in multiple variables, which would require several iterations to complete the imputation when more than one variable is missing. The main limitations of this approach are that the model parameters may be suboptimal and that it requires to train and deploy several models, one for every combination of conditions. Moreover, if the conditions of the gap are not similar to any trained model (e.g. a different EC site), there can be a significant decrease in performance.

A variation of the approach above is to use the similarity between the training and inference condition to select a set of candidate models, instead of only one. Then each candidate model will be used to impute artificial gaps created with conditions similar to the missing data, and the model with the smallest error will be eventually employed to impute the gap. This method is better at finding the best parameters for new conditions, but it requires additional computational resources to test the different candidate models.

The last method is to fine tune the model to the conditions of each gap. This provides the best performance and flexibility at the cost of increased computational cost. Moreover, this requires the development of criteria to automatically stop the training. Furthermore, the training difficulties of the KF may prevent the parameter optimization.

The best KF deployment approach depends on the specific application (e.g. imputation of a single site or multiple sites).

4.3 Future outlook

Kalman Filter improvements This work builds the foundations for applying a KF based method for imputing meteorological variables, but further work is needed to develop this approach.

As described in the previous section, there are two main issues that would prevent the use of the KF in a real world scenario. The numerical instability of the current KF implementation limits the gaps to 15 hours, when all variables are missing. More research is needed to develop a suitable formulation of a square root smoother, which the available literature suggests that it is possible to derive [44, 38]. The negative predictions of `SW_IN` at night can be solved by using the KF to predict `SW_IN` during the day, and then replacing all the predictions with zero for the night. The exact time of the sunrise and sunset, and so nighttime, can be easily computed from the day of the year and the geographic coordinates.

The European Centre for Medium-Range Weather Forecasts recently released two new weather reanalysis datasets: ERA5 in 2020 [23] and ERA-5 Land in 2021 [36], which supersede the ERA-Interim dataset. The ERA5-Land dataset covers only the continents, but has a much higher spatial resolution (9 km instead of 80 km) and higher temporal resolution (1 hour instead of 3 hours) compared to ERA-I. Therefore, the prediction of the KF can be improved by using ERA5-Land data instead of ERA-I as the control variable.

My analysis suggests that initial parameters have significant influence on the final parameters and thus the performance of the model. There are several approaches to initialize a KF [17], and a robust comparison between the different methods may reveal a better initialization strategy than the linear trend model.

The current KF implementation is limited to model only linear relationship between states and between the control and the state. There exist non-linear versions of the KF, such as the Extended Kalman Filter and the Unscented Kalman Filter [14], that approximate non-linear relationship between successive states. The relation between the ERA-I data and the observed variables is also non-linear [49]. The KF does not require linear relationship between the state and the control, so any arbitrary transformation could be applied. For instance, the use of Neural Network could be assessed. This would also integrate well with the implementation in PyTorch of the KF.

One parameter of the KF is the observation covariance (R). Currently, this parameter is estimated by the general optimization procedure, but the actual value of the observations' noise can be derived from the instrument accuracy, as the KF is a probabilistic model. This could prevent overly confident predictions.

One of the advantages of the iterative nature of the KF formulation is the ability to change the parameters between time steps. This opens the possibility to utilize a model to predict the parameters of the KF depending on the meteorological conditions. This should overcome the necessity to fine-tune the KF parameters to each condition. However, this may not be possible due to the difficulty of learning the parameters of the KF in the first place.

Model Evaluation This study provides a first evaluation of the imputation performance of the KF, but a more in-depth analysis iss necessary to understand the real impact of applying the KF.

The first aspect to consider is that the meteorological data is not missing completely at random. Further research would be needed, but it is reasonable to assume that there is a correlation between the gaps of different variables and that the time of the year has an impact on data availability. For instance, one reason that can contribute to patterns in the missing data is that `TA` and `VPD` are often measured from the same sensor [1, 46], which increases the likelihood of both variables missing at the same time. Another scenario is a station that uses solar panels, which is more likely to have power failure during winter. A robust assessment of the imputation performance should have the pattern in artificial gaps reflecting real-world conditions.

Another aspect is to test the imputation performance using observations from different sites. EC sites are located all over the globe [39] and the difference between climate conditions results in different patterns in the temporal autocorrelation and inter-variable correlation, which affects in turn the performance of the KF.

The metric used for the evaluation is also important. The RMSE measures the average distance between the predictions and the observations, but does not compare other characteristics of the time series such as the variance, the shape or the presence of a time-shift [21]. In this study, the visual inspection of the time series was used to assess the imputation quality. This method is, however, time-consuming and is quantitative. Additional metrics, such as DILATE [21], that measure other aspect of the predictions can improve the understanding of the quality of the imputation.

Finally, the performance of the KF imputation can be evaluated by utilizing Land Surface Models (LSM). A time series with artificial gaps could be imputed using both the KF and state-of-the-art methods and then by measuring the difference in the error of the LSM compared to the original time series. This would allow to directly measure the actual impact of the improved imputation of meteorological variables.

5 Conclusions

The imputation of meteorological time series is necessary for EC applications. The Kalman Filter (KF) outperforms the state-of-the-art methods (ERA-I and MDS) for all variables but precipitation. The strengths of the KF are: the ability to combine different imputation approaches; the effective use of the variable temporal autocorrelation; the quantification of the predictions' uncertainty.

The imputation performance of the KF depends on the missing variable: the smallest error is for air temperature, air pressure, soil temperature and vapor pressure deficit. The air temperature is the variable with the biggest error reduction compared to state-of-the-art methods. The KF has an intermediate performance for the incoming shortwave, longwave radiation, and the wind speed. For the wind speed, the KF is limited in modelling the short term variability. The KF, like the other methods, has a poor imputation performance for the precipitation, whose unique characteristics do not make the KF a suitable method.

The current implementation of the KF is affected by numerical instability issues, which limits the model to 15 hours long gaps when all variables are missing, and at night predicts physically impossible values of the shortwave radiation. Nonetheless, both issues can be resolved by further research. However, I also identified limitations inherit in the Kalman Filter approach: the necessity to fine-tune the parameters and the difficulties to train the model. These issues do not prevent the use of the KF, but they may result in suboptimal predictions and in complex deployment setups.

This work shows the potential of applying the KF for EC meteorological time series imputation, however additional work is needed to further develop the imputation method and to better assess its performance.

References

[1] *Associated ICOS Ecosystem Station Labelling Report - Hainich.* 2020. URL: https://data.icos-cp.eu/objects/_tFsWRgQcO7FkfvOqOOqIC8H (visited on 02/18/2023).

[2] Marc Aubinet, Timo Vesala, and Dario Papale, eds. *Eddy Covariance: A Practical Guide to Measurement and Data Analysis.* Dordrecht: Springer Netherlands, 2012. ISBN: 978-94-007-2350-4 978-94-007-2351-1. DOI: 10.1007/978-94-007-2351-1. URL: https://link.springer.com/10.1007/978-94-007-2351-1 (visited on 02/10/2023).

[3] M. Balzarolo et al. "Evaluating the Potential of Large-Scale Simulations to Predict Carbon Fluxes of Terrestrial Ecosystems over a European Eddy Covariance Network". In: *Biogeosciences* 11.10 (May 20, 2014), pp. 2661–2678. ISSN: 1726-4170. DOI: 10.5194/bg-11-2661-2014. URL: https://bg.copernicus.org/articles/11/2661/2014/ (visited on 02/10/2023).

[4] Simon Besnard et al. "Quantifying the Effect of Forest Age in Annual Net Forest Carbon Balance". In: *Environmental Research Letters* 13.12 (Dec. 2018), p. 124018. ISSN: 1748-9326. DOI: 10.1088/1748-9326/aaeaeb. URL: https://dx.doi.org/10.1088/1748-9326/aaeaeb (visited on 02/24/2023).

[5] Gerald J. Bierman and Catherine L. Thornton. "Numerical Comparison of Kalman Filter Algorithms: Orbit Determination Case Study". In: *Automatica* 13.1 (Jan. 1, 1977), pp. 23–35. ISSN: 0005-1098. DOI: 10.1016/0005-1098(77)90006-1. URL: https://www.sciencedirect.com/science/article/pii/0005109877900061 (visited on 01/12/2023).

[6] Bishop. *Pattern Recognition and Machine Learning.* 2006.

[7] Gordon Bonan. *Climate Change and Terrestrial Ecosystem Modeling.* Cambridge University Press, 2019.

[8] Gordon B. Bonan et al. "Improving Canopy Processes in the Community Land Model Version 4 (CLM4) Using Global Flux Fields Empirically Inferred from FLUXNET Data". In: *Journal of Geophysical Research: Biogeosciences* 116.G2 (2011). ISSN: 2156-2202. DOI: 10.1029/2010JG001593. URL: https://onlinelibrary.wiley.com/doi/abs/10.1029/2010JG001593 (visited on 02/10/2023).

[9] Stef van Buuren and Karin Groothuis-Oudshoorn. "Mice: Multivariate Imputation by Chained Equations in R". In: *Journal of Statistical Software* 45 (Dec. 12, 2011), pp. 1–67. ISSN: 1548-7660. DOI: 10.18637/jss.v045.i03. URL: https://doi.org/10.18637/jss.v045.i03 (visited on 02/18/2023).

[10] Wei Cao et al. "BRITS: Bidirectional Recurrent Imputation for Time Series". In: (), p. 11.

[11] NEAL A. CARLSON. "Fast Triangular Formulation of the Square Root Filter." In: *AIAA Journal* 11.9 (1973), pp. 1259–1265. ISSN: 0001-1452. DOI: 10.2514/3.6907. URL: https://doi.org/10.2514/3.6907 (visited on 02/15/2023).

[12] Benedict D. Chivers et al. "Imputation of Missing Sub-Hourly Precipitation Data in a Large Sensor Network: A Machine Learning Approach". In: *Journal of Hydrology* 588 (Sept. 1, 2020), p. 125126. ISSN: 0022-1694. DOI: 10.1016/j.jhydrol.2020.125126. URL: https://www.sciencedirect.com/science/article/pii/S0022169420305862 (visited on 02/21/2023).

[13] Rafaela Lisboa Costa et al. "Gap Filling and Quality Control Applied to Meteorological Variables Measured in the Northeast Region of Brazil". In: *Atmosphere* 12.10 (10 Oct. 2021), p. 1278. ISSN: 2073-4433. DOI: 10.3390/atmos12101278. URL: https://www.mdpi.com/2073-4433/12/10/1278 (visited on 05/14/2022).

[14] Dan Simon. *Optimal State Estimation Kalman, H and Nonlinear Approaches*. 2006. ISBN: 100-47 1-70858-5.

[15] D. P. Dee et al. “The ERA-Interim Reanalysis: Configuration and Performance of the Data Assimilation System”. In: *Quarterly Journal of the Royal Meteorological Society* 137.656 (2011), pp. 553–597. ISSN: 1477-870X. DOI: 10.1002/qj.828. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/qj.828 (visited on 02/24/2023).

[16] Wenjie Du, David Cote, and Yan Liu. *SAITS: Self-Attention-based Imputation for Time Series*. May 9, 2022. arXiv: arXiv:2202.08516. URL: http://arxiv.org/abs/2202.08516 (visited on 05/14/2022). preprint.

[17] James Durbin and Siem Jan Koopman. *Time Series Analysis by State Space Methods*. Mar. 2, 2012. URL: https://doi.org/10.1093/acprof:oso/9780199641178.001.0001.

[18] James Durbin and Siem Jan Koopman. *Time Series Analysis by State Space Methods: Second Edition*. Oxford University Press, May 3, 2012. ISBN: 978-0-19-964117-8. DOI: 10.1093/acprof:oso/9780199641178.001.0001. URL: http://www.oxfordscholarship.com/view/10.1093/acprof:oso/9780199641178.001.0001/acprof-9780199641178 (visited on 11/13/2022).

[19] Chenguang Fang and Chen Wang. “Time Series Data Imputation: A Survey on Deep Learning Approaches”. Nov. 23, 2020. arXiv: 2011.11347 [cs]. URL: http://arxiv.org/abs/2011.11347 (visited on 05/12/2022).

[20] Andrew D. Friend et al. “FLUXNET and Modelling the Global Carbon Cycle”. In: *Global Change Biology* 13.3 (2007), pp. 610–633. ISSN: 1365-2486. DOI: 10.1111/j.1365-2486.2006.01223.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2486.2006.01223.x (visited on 02/10/2023).

[21] Vincent LE Guen and Nicolas Thome. “Shape and Time Distortion Loss for Training Deep Time Series Forecasting Models”. In: (), p. 13.

[22] Philipp Hennig. *Probabilistic Machine Learning*. lecture course. University of Tuebingen, 2020.

[23] Hans Hersbach et al. “The ERA5 Global Reanalysis”. In: *Quarterly Journal of the Royal Meteorological Society* 146.730 (2020), pp. 1999–2049. ISSN: 1477-870X. DOI: 10.1002/qj.3803. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/qj.3803 (visited on 02/22/2023).

[24] Jeremy Howard and Sylvain Gugger. “Fastai: A Layered API for Deep Learning”. In: *Information* 11.2 (Feb. 16, 2020), p. 108. ISSN: 2078-2489. DOI: 10.3390/info11020108. arXiv: 2002.04688 [cs, stat]. URL: http://arxiv.org/abs/2002.04688 (visited on 02/28/2023).

[25] Peter Isaac et al. “OzFlux Data: Network Integration from Collection to Curation”. In: *Biogeosciences* 14.12 (June 19, 2017), pp. 2903–2928. ISSN: 1726-4170. DOI: 10.5194/bg-14-2903-2017. URL: https://bg.copernicus.org/articles/14/2903/2017/ (visited on 05/12/2022).

[26] Xin Jing et al. “A Multi-imputation Method to Deal With Hydro-Meteorological Missing Values by Integrating Chain Equations and Random Forest”. In: *Water Resources Management* 36.4 (Mar. 1, 2022), pp. 1159–1173. ISSN: 1573-1650. DOI: 10.1007/s11269-021-03037-5. URL: https://doi.org/10.1007/s11269-021-03037-5 (visited on 02/18/2023).

[27] P. Kaminski, A. Bryson, and S. Schmidt. “Discrete Square Root Filtering: A Survey of Current Techniques”. In: *IEEE Transactions on Automatic Control* 16.6 (Dec. 1971), pp. 727–736. ISSN: 1558-2523. DOI: 10.1109/TAC.1971.1099816.

[28] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization.* Jan. 29, 2017. DOI: 10.48550/arXiv.1412.6980. arXiv: arXiv:1412.6980. URL: http://arxiv.org/abs/1412.6980 (visited on 02/26/2023). preprint.

[29] K. Kramer et al. “Evaluation of Six Process-Based Forest Growth Models Using Eddy-Covariance Measurements of CO2 and H2O Fluxes at Six Forest Sites in Europe”. In: *Global Change Biology* 8.3 (2002), pp. 213–230. ISSN: 1365-2486. DOI: 10.1046/j.1365-2486.2002.00471.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1046/j.1365-2486.2002.00471.x (visited on 01/18/2023).

[30] Miguel D. Mahecha et al. “Detecting Impacts of Extreme Events with Ecological in Situ Monitoring Networks”. In: *Biogeosciences* 14.18 (Sept. 25, 2017), pp. 4255–4277. ISSN: 1726-4170. DOI: 10.5194/bg-14-4255-2017. URL: https://bg.copernicus.org/articles/14/4255/2017/ (visited on 02/24/2023).

[31] Utkarsh Mital et al. “Sequential Imputation of Missing Spatio-Temporal Precipitation Data Using Random Forests”. In: *Frontiers in Water* 2 (2020). ISSN: 2624-9375. URL: https://www.frontiersin.org/articles/10.3389/frwa.2020.00020 (visited on 02/21/2023).

[32] Antje M. Moffat et al. “Comprehensive Comparison of Gap-Filling Techniques for Eddy Covariance Net Carbon Fluxes”. In: *Agricultural and Forest Meteorology* 147.3 (Dec. 10, 2007), pp. 209–232. ISSN: 0168-1923. DOI: 10.1016/j.agrformet.2007.08.011. URL: https://www.sciencedirect.com/science/article/pii/S016819230700216X (visited on 05/12/2022).

[33] Mohinder S. Grewal and Angus P. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB, Second Edition.* 2001. ISBN: 0-471-26638-8.

[34] Steffen Moritz. “Comparison of Different Methods for Univariate Time Series Imputation in R”. In: ().

[35] Steffen Moritz and Thomas Bartz-Beielstein. “The R Journal: imputeTS: Time Series Missing Value Imputation in R”. In: *The R Journal* 9.1 (May 10, 2017), pp. 207–218. ISSN: 2073-4859. DOI: 10.32614/RJ-2017-009. URL: https://doi.org/10.32614/RJ-2017-009/ (visited on 02/24/2023).

[36] Joaquín Muñoz-Sabater et al. “ERA5-Land: A State-of-the-Art Global Reanalysis Dataset for Land Applications”. In: *Earth System Science Data* 13.9 (Sept. 7, 2021), pp. 4349–4383. ISSN: 1866-3508. DOI: 10.5194/essd-13-4349-2021. URL: https://essd.copernicus.org/articles/13/4349/2021/ (visited on 02/22/2023).

[37] Dario Papale. “Ideas and Perspectives: Enhancing the Impact of the FLUXNET Network of Eddy Covariance Sites”. In: *Biogeosciences* 17.22 (Nov. 17, 2020), pp. 5587–5598. ISSN: 1726-4189. DOI: 10.5194/bg-17-5587-2020. URL: https://bg.copernicus.org/articles/17/5587/2020/ (visited on 01/18/2023).

[38] PooGyeon Park and T. Kailath. “New Square-Root Smoothing Algorithms”. In: *IEEE Transactions on Automatic Control* 41.5 (May 1996), pp. 727–732. ISSN: 1558-2523. DOI: 10.1109/9.489212.

[39] Gilberto Pastorello et al. “The FLUXNET2015 Dataset and the ONEFlux Processing Pipeline for Eddy Covariance Data”. In: *Scientific Data* 7.1 (1 July 9, 2020), p. 225. ISSN: 2052-4463. DOI: 10.1038/s41597-020-0534-3. URL: https://www.nature.com/articles/s41597-020-0534-3 (visited on 05/12/2022).

[40] Adam Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[41] JAMES POTTER and ROBERT STERN. “STATISTICAL FILTERING OF SPACE NAVIGATION MEASUREMENTS”. In: *Guidance and Control Conference*. American Institute of Aeronautics and Astronautics, 1963. DOI: 10.2514/6.1963-333. URL: https://arc.aiaa.org/doi/abs/10.2514/6.1963-333 (visited on 01/13/2023).

[42] H. E. RAUCH, F. TUNG, and C. T. STRIEBEL. “Maximum Likelihood Estimates of Linear Dynamic Systems”. In: *AIAA Journal* 3.8 (1965), pp. 1445–1450. ISSN: 0001-1452. DOI: 10.2514/3.3166. URL: https://doi.org/10.2514/3.3166 (visited on 02/19/2023).

[43] Markus Reichstein et al. “On the Separation of Net Ecosystem Exchange into Assimilation and Ecosystem Respiration: Review and Improved Algorithm”. In: *Global Change Biology* 11.9 (2005), pp. 1424–1439. ISSN: 1365-2486. DOI: 10.1111/j.1365-2486.2005.001002.x. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1365-2486.2005.001002.x (visited on 05/12/2022).

[44] Mark G. Rutten. “Square-Root Unscented Filtering and Smoothing”. In: *2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*. 2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing. Apr. 2013, pp. 294–299. DOI: 10.1109/ISSNIP.2013.6529805.

[45] Arvind Satyanarayan et al. “Vega-Lite: A Grammar of Interactive Graphics”. In: *IEEE transactions on visualization and computer graphics* 23.1 (2017), pp. 341–350.

[46] *Specification - Vaisala HMP3 General Purpose Humidity and Temperature Probe*. URL: https://docs.vaisala.com/v/u/B211826EN-C/en-US (visited on 02/18/2023).

[47] *Statsmodels.Tsa.Statespace.Kalman_filter.KalmanFilter — Statsmodels*. URL: https://www.statsmodels.org/dev/generated/statsmodels.tsa.statespace.kalman_filter.KalmanFilter.html#statsmodels.tsa.statespace.kalman_filter.KalmanFilter (visited on 02/19/2023).

[48] Jacob VanderPlas et al. “Altair: Interactive Statistical Visualizations for Python”. In: *Journal of Open Source Software* 3.32 (2018), p. 1057. DOI: 10.21105/joss.01057. URL: https://doi.org/10.21105/joss.01057.

[49] N. Vuichard and D. Papale. “Filling the Gaps in Meteorological Continuous Data Measured at FLUXNET Sites with ERA-Interim Reanalysis”. In: *Earth System Science Data* 7.2 (July 13, 2015), pp. 157–171. ISSN: 1866-3508. DOI: 10.5194/essd-7-157-2015. URL: https://essd.copernicus.org/articles/7/157/2015/ (visited on 05/11/2022).

[50] Thomas Wutzler et al. “Basic and Extensible Post-Processing of Eddy Covariance Flux Data with REddyProc”. In: *Biogeosciences* 15.16 (Aug. 23, 2018), pp. 5015–5030. ISSN: 1726-4170. DOI: 10.5194/bg-15-5015-2018. URL: https://bg.copernicus.org/articles/15/5015/2018/ (visited on 05/15/2022).

[51] Yifan Zhang and Peter J. Thorburn. “A Dual-Head Attention Model for Time Series Data Imputation”. In: *Computers and Electronics in Agriculture* 189 (Oct. 1, 2021), p. 106377. ISSN: 0168-1699. DOI: 10.1016/j.compag.2021.106377. URL: https://www.sciencedirect.com/science/article/pii/S016816992100394X (visited on 06/15/2022).

[52] Y. Zhao et al. “How Errors on Meteorological Variables Impact Simulated Ecosystem Fluxes: A Case Study for Six French Sites”. In: *Biogeosciences* 9.7 (July 11, 2012), pp. 2537–2564. ISSN: 1726-4170. DOI: 10.5194/bg-9-2537-2012. URL: https://bg.copernicus.org/articles/9/2537/2012/ (visited on 02/13/2023).

A Additional Results

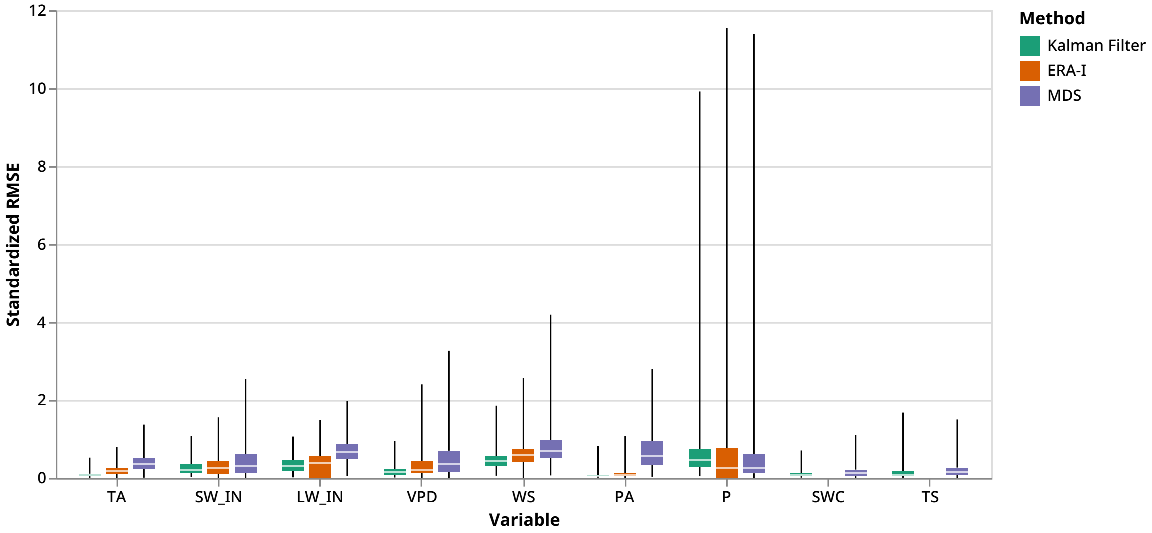


Figure A.1: Box plot to compare Standardized Root Mean Square Error(RMSE) for each variable between the different methods: Kalman Filter and the state-of-the-art methods ERA and MDS. The same data from Figure 5 has been aggregated for all gap lengths. The extent of the box plot vertical lines represent the maximum and minimum value of a gap RMSE.

Table A.1: Imputation performance of the Kalman filter in comparison to the state-of-the-art methods: ERA-Interim (ERA-I) and Marginal Distribution Sampling (MDS), using mean and standard deviation of the *Root Mean Square Error* (RMSE). The best method for each gap length is highlighted in bold. For each combination of gap length and variable, 500 artificial gaps were created.

		Kalman Filter		ERA-I		MDS	
	Stand. RMSE	mean	std	mean	std	mean	std
Variable	Gap						
TA	6 h	**0.051**	0.033	0.170	0.126	0.342	0.239
	12 h	**0.077**	0.051	0.186	0.114	0.371	0.221
	1 day (24 h)	**0.094**	0.046	0.193	0.101	0.380	0.203
	1 week (168 h)	**0.129**	0.056	0.221	0.081	0.477	0.166
SW_IN	6 h	**0.219**	0.198	0.242	0.325	0.311	0.419
	12 h	**0.236**	0.166	0.266	0.244	0.340	0.338
	1 day (24 h)	**0.277**	0.147	0.323	0.201	0.425	0.292
	1 week (168 h)	**0.302**	0.126	0.344	0.171	0.526	0.263
LW_IN	6 h	**0.260**	0.184	0.329	0.310	0.636	0.358
	12 h	**0.320**	0.184	0.352	0.300	0.669	0.321
	1 day (24 h)	0.348	0.187	**0.336**	0.291	0.706	0.296
	1 week (168 h)	0.407	0.153	**0.390**	0.265	0.785	0.211
VPD	6 h	**0.098**	0.083	0.297	0.354	0.477	0.492
	12 h	**0.151**	0.116	0.290	0.295	0.489	0.480
	1 day (24 h)	**0.189**	0.115	0.286	0.236	0.438	0.367
	1 week (168 h)	**0.258**	0.145	0.380	0.258	0.609	0.450
WS	6 h	**0.379**	0.195	0.561	0.313	0.699	0.482
	12 h	**0.440**	0.216	0.588	0.323	0.776	0.490
	1 day (24 h)	**0.493**	0.211	0.584	0.275	0.785	0.374
	1 week (168 h)	**0.585**	0.223	0.670	0.214	0.920	0.379
PA	6 h	**0.053**	0.040	0.088	0.072	0.621	0.516
	12 h	**0.062**	0.049	0.090	0.068	0.659	0.500
	1 day (24 h)	**0.070**	0.045	0.092	0.060	0.651	0.473
	1 week (168 h)	**0.078**	0.056	0.098	0.063	0.904	0.449
P	6 h	0.478	0.978	**0.404**	1.126	0.420	1.090
	12 h	0.638	1.054	0.495	1.060	**0.465**	1.004
	1 day (24 h)	0.736	0.905	0.591	1.029	**0.566**	0.946
	1 week (168 h)	0.856	0.620	0.795	0.720	**0.767**	0.705
SWC	6 h	**0.057**	0.055	-	-	0.147	0.175
	12 h	**0.075**	0.053	-	-	0.143	0.148
	1 day (24 h)	**0.087**	0.072	-	-	0.152	0.165
	1 week (168 h)	**0.168**	0.106	-	-	0.219	0.167
TS	6 h	**0.060**	0.076	-	-	0.169	0.157
	12 h	**0.094**	0.139	-	-	0.177	0.155
	1 day (24 h)	**0.139**	0.151	-	-	0.191	0.151
	1 week (168 h)	0.293	0.190	-	-	**0.254**	0.135

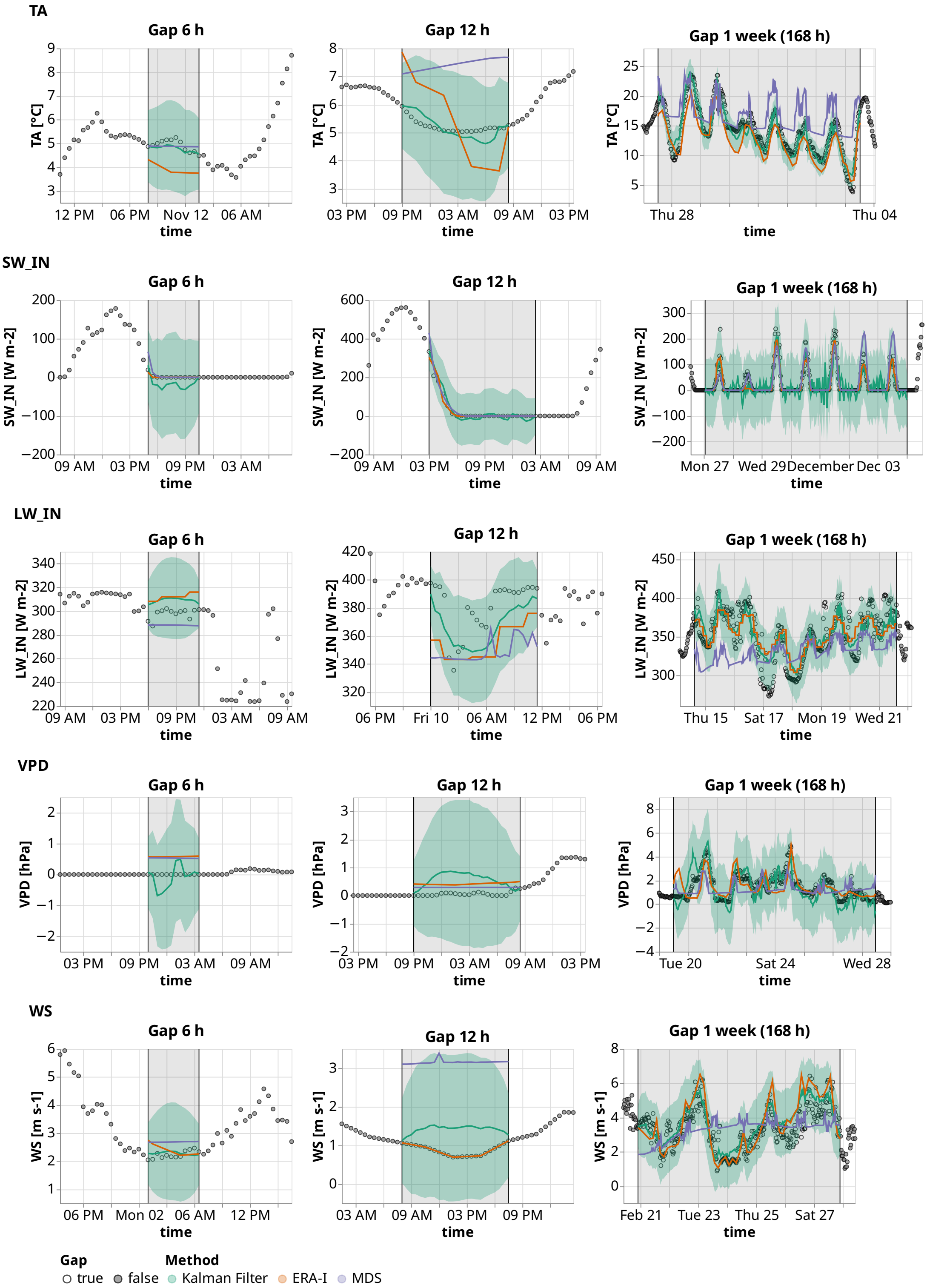


Figure A.2: Second time series to visualize the imputation of TA, SW_IN, LW_IN, VPD, WS using different methods: Kalman Filter (KF), ERA-Interim (ERA-I) and Marginal Distribution Sampling (MDS). For each variable, three random artificial gap (length 6 hours, 12 hours, 1 week) are imputed using the three methods: Kalman Filter (green), ERA-I (orange), MDS (purple). For the KF the shaded area shows the uncertainty of the prediction ±2σ (standard deviation). The grey shaded area and the vertical black lines delimit the artificial gaps, where the observations are not available to the model but are used to assess the imputation performance. The ERA-I prediction is the control variable of the KF. The KF model has been fine-tuned to each variable (*KF-⟨var⟩-Sin-6_336*).

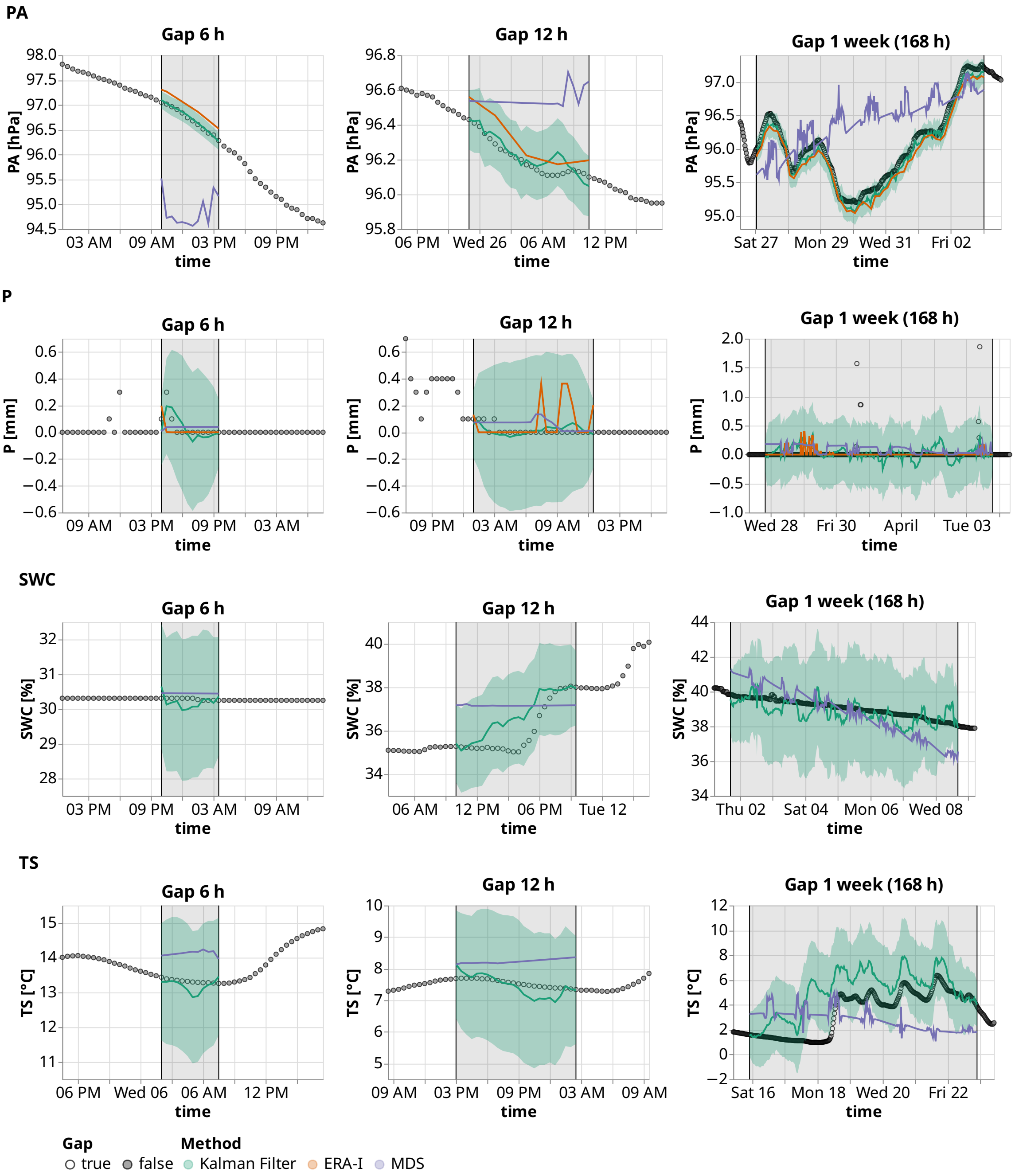


Figure A.3: Second time series to visualize the imputation of PA, P, SWC, TS using different methods: Kalman Filter (KF), ERA-Interim (ERA-I) and Marginal Distribution Sampling (MDS). For each variable, three random artificial gap (length 6 hours, 12 hours, 1 week) are imputed using the three methods: Kalman Filter (green), ERA-I (orange), MDS (purple). For the KF the shaded area shows the uncertainty of the prediction $\pm 2\sigma$ (standard deviation). The grey shaded area and the vertical black lines delimit the artificial gaps, where the observations are not available to the model but are used to assess the imputation performance. The ERA-I prediction is the control variable of the KF. The KF model has been fine-tuned to each variable (*KF-⟨var⟩-Sin-6_336*).

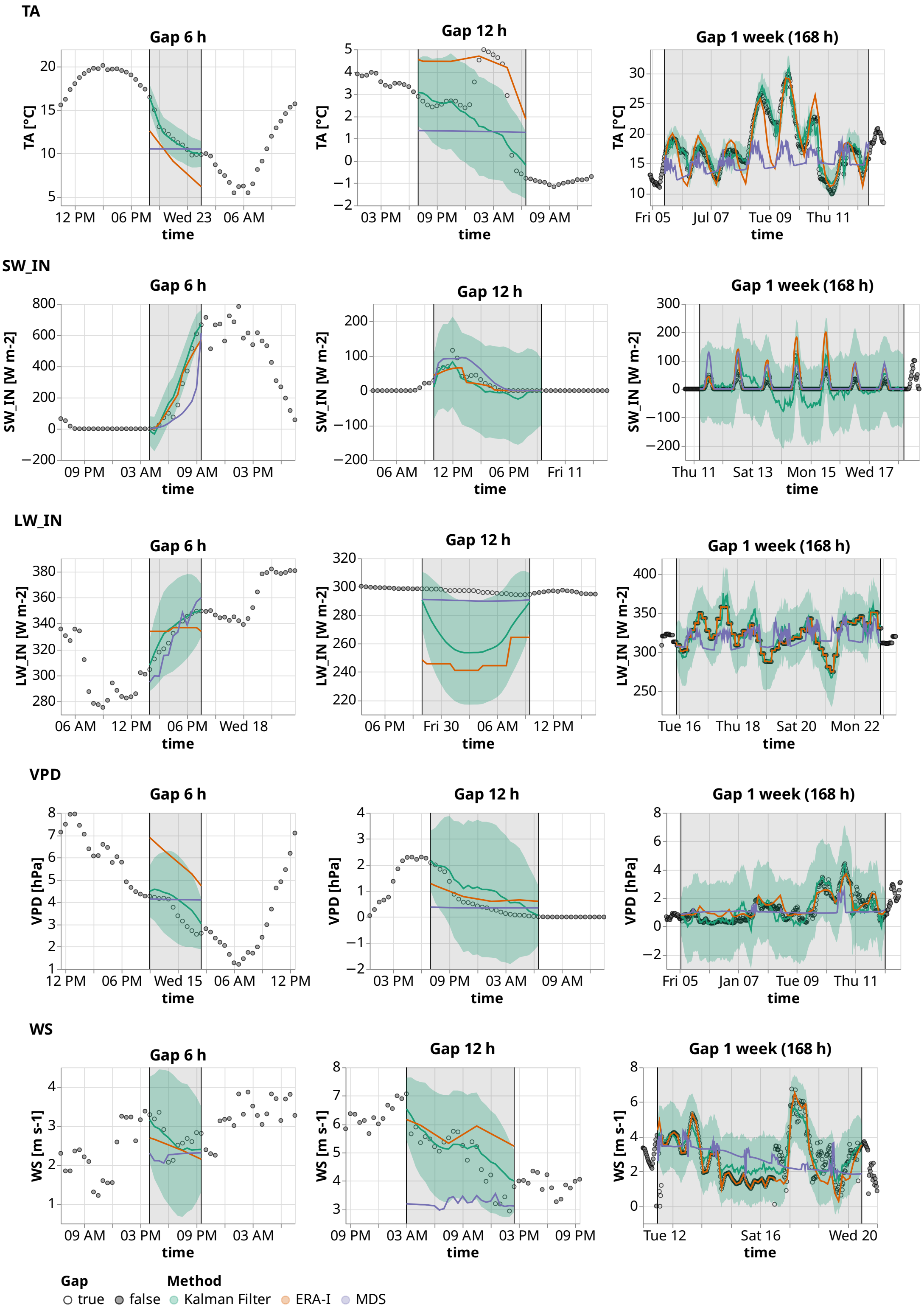


Figure A.4: Third time series to visualize the imputation of `TA`, `SW_IN`, `LW_IN`, `VPD`, `WS` using different methods: Kalman Filter (KF), ERA-Interim (ERA-I) and Marginal Distribution Sampling (MDS). For each variable, three random artificial gap (length 6 hours, 12 hours, 1 week) are imputed using the three methods: Kalman Filter (green), ERA-I (orange), MDS (purple). For the KF the shaded area shows the uncertainty of the prediction ±2σ (standard deviation). The grey shaded area and the vertical black lines delimit the artificial gaps, where the observations are not available to the model but are used to assess the imputation performance. The ERA-I prediction is the control variable of the KF. The KF model has been fine-tuned to each variable (*KF-⟨var⟩-Sin-6_336*).

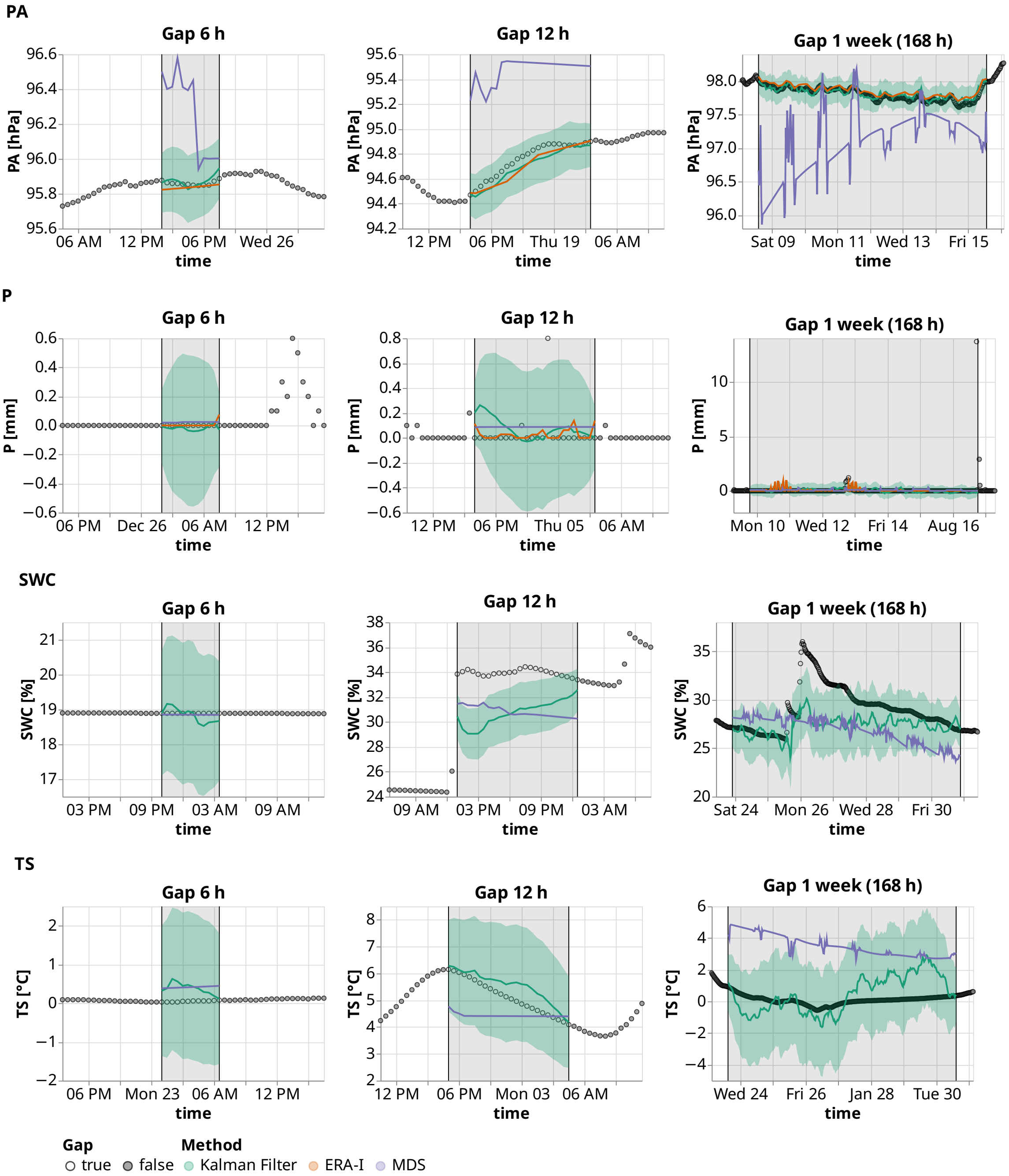


Figure A.5: Third time series to visualize the imputation of PA, P, SWC, TS using different methods: Kalman Filter (KF), ERA-Interim (ERA-I) and Marginal Distribution Sampling (MDS). For each variable, three random artificial gap (length 6 hours, 12 hours, 1 week) are imputed using the three methods: Kalman Filter (green), ERA-I (orange), MDS (purple). For the KF the shaded area shows the uncertainty of the prediction ±2σ (standard deviation). The grey shaded area and the vertical black lines delimit the artificial gaps, where the observations are not available to the model but are used to assess the imputation performance. The ERA-I prediction is the control variable of the KF. The KF model has been fine-tuned to each variable (*KF-⟨var⟩-Sin-6_336*).

Table A.2: Imputation performance of the KF in comparison to the state-of-the-art methods: ERA-Interim (ERA-I) and Marginal Distribution Sampling (MDS), using mean and standard deviation of the *Root Mean Square Error* (RMSE). The best method for each gap length is highlighted in bold. For each combination of gap length and variable, 500 artificial gaps were created.

	Gap	1 h	3 h	6 h	12 h	1 day (24 h)	2 days (48 h)	3 days (72 h)	1 week (168 h)
Variable	Stand. RMSE								
TA	mean	0.025	0.034	0.055	0.078	0.107	0.116	0.119	0.125
	std	0.023	0.022	0.038	0.048	0.059	0.056	0.052	0.043
SW_IN	mean	0.141	0.185	0.219	0.242	0.283	0.279	0.286	0.295
	std	0.165	0.181	0.192	0.163	0.139	0.130	0.125	0.120
LW_IN	mean	0.122	0.200	0.251	0.356	0.364	0.401	0.400	0.426
	std	0.160	0.172	0.182	0.227	0.189	0.195	0.186	0.164
VPD	mean	0.044	0.072	0.106	0.186	0.203	0.227	0.238	0.259
	std	0.050	0.093	0.093	0.177	0.136	0.142	0.133	0.149
WS	mean	0.237	0.305	0.372	0.473	0.506	0.551	0.531	0.565
	std	0.198	0.193	0.202	0.274	0.285	0.239	0.201	0.196
PA	mean	0.027	0.038	0.055	0.064	0.068	0.075	0.075	0.080
	std	0.027	0.030	0.066	0.040	0.051	0.059	0.055	0.060
P	mean	0.246	0.550	0.443	0.722	0.694	0.729	0.851	0.908
	std	0.764	1.822	0.525	1.352	0.783	0.620	0.750	0.583
SWC	mean	0.020	0.039	0.054	0.068	0.088	0.099	0.113	0.166
	std	0.016	0.035	0.036	0.042	0.062	0.072	0.086	0.099
TS	mean	0.026	0.044	0.067	0.105	0.130	0.186	0.210	0.316
	std	0.026	0.043	0.071	0.145	0.119	0.148	0.140	0.218

Table A.3: Comparison of the KF performance between a model that uses control variables (*KF-Gen-Sin-6_336*) and models that do not use control variables (*KF-Gen-Sin-6_336-No_Contr*). The table displays the mean, the standard deviation (std) and the standard error (se) of the *Root Mean Square Error* (RMSE). In addition the difference (diff.) between the mean of two models is shown. The best method for each gap length is highlighted in bold. For each combination of gap length and variable, 500 artificial gaps were created.

	type		Use Control			No Control		
	Stand. RMSE	mean	std	se	mean	std	se	diff.
Variable	Gap							
TA	6 h	**0.092**	0.056	0.002	0.095	0.059	0.003	0.002
	12 h	**0.118**	0.074	0.003	0.161	0.107	0.005	0.042
	1 day (24 h)	**0.147**	0.077	0.003	0.233	0.159	0.007	0.086
	1 week (168 h)	**0.174**	0.064	0.003	0.294	0.148	0.007	0.120
SW_IN	6 h	**0.256**	0.165	0.007	0.331	0.271	0.012	0.075
	12 h	**0.315**	0.165	0.007	0.519	0.320	0.014	0.204
	1 day (24 h)	**0.339**	0.129	0.006	0.609	0.277	0.012	0.270
	1 week (168 h)	**0.358**	0.103	0.005	0.693	0.248	0.011	0.335
LW_IN	6 h	**0.299**	0.201	0.009	0.330	0.185	0.008	0.031
	12 h	**0.368**	0.197	0.009	0.486	0.193	0.009	0.118
	1 day (24 h)	**0.400**	0.176	0.008	0.561	0.187	0.008	0.161
	1 week (168 h)	**0.466**	0.141	0.006	0.666	0.161	0.007	0.200
VPD	6 h	**0.138**	0.105	0.005	0.179	0.139	0.006	0.041
	12 h	**0.227**	0.184	0.008	0.321	0.263	0.012	0.094
	1 day (24 h)	**0.273**	0.207	0.009	0.408	0.297	0.013	0.135
	1 week (168 h)	**0.325**	0.154	0.007	0.499	0.252	0.011	0.174
WS	6 h	0.572	0.523	0.023	**0.430**	0.262	0.012	-0.142
	12 h	**0.660**	0.409	0.018	0.680	0.392	0.018	0.020
	1 day (24 h)	**0.676**	0.374	0.017	0.767	0.430	0.019	0.091
	1 week (168 h)	**0.782**	0.265	0.012	0.889	0.318	0.014	0.107
PA	6 h	**0.099**	0.066	0.003	0.145	0.105	0.005	0.045
	12 h	**0.116**	0.061	0.003	0.339	0.210	0.009	0.222
	1 day (24 h)	**0.137**	0.076	0.003	0.563	0.379	0.017	0.426
	1 week (168 h)	**0.159**	0.068	0.003	0.880	0.440	0.020	0.721
P	6 h	0.542	0.991	0.044	**0.508**	0.974	0.044	-0.035
	12 h	0.627	1.035	0.046	**0.605**	1.025	0.046	-0.022
	1 day (24 h)	0.628	0.575	0.026	**0.614**	0.584	0.026	-0.014
	1 week (168 h)	0.881	0.573	0.026	**0.861**	0.597	0.027	-0.020
SWC	6 h	0.152	0.107	0.005	**0.151**	0.141	0.006	-0.001
	12 h	**0.245**	0.154	0.007	0.305	0.234	0.010	0.061
	1 day (24 h)	**0.393**	0.223	0.010	0.538	0.338	0.015	0.145
	1 week (168 h)	**0.719**	0.314	0.014	0.842	0.431	0.019	0.123
TS	6 h	0.168	0.115	0.005	**0.119**	0.076	0.003	-0.049
	12 h	0.226	0.135	0.006	**0.203**	0.133	0.006	-0.023
	1 day (24 h)	0.286	0.152	0.007	**0.285**	0.174	0.008	-0.001
	1 week (168 h)	0.362	0.176	0.008	**0.359**	0.191	0.009	-0.003

Table A.4: Comparison of the KF performance of a gap when only one variable missing and all other variables are missing. The model used for imputation is always *KF-Gen-Multi-6_336*. The table displays the mean, the standard deviation (std) and the standard error (se) of the *Root Mean Square Error* (RMSE). In addition the difference (diff.) between the mean of two models is shown. The best method for each gap length is highlighted in bold. For each combination of gap length and variable, 500 artificial gaps were created.

	Gap	Only one var			All variables			
	Stand. RMSE	mean	std	se	mean	std	se	diff.
Variable	Gap							
TA	3 h	**0.029**	0.026	0.001	0.048	0.040	0.002	0.019
	6 h	**0.043**	0.032	0.001	0.081	0.063	0.003	0.038
	12 h	**0.071**	0.050	0.002	0.124	0.097	0.004	0.053
	15 h	**0.079**	0.054	0.002	0.136	0.088	0.004	0.057
SW_IN	3 h	0.199	0.205	0.009	**0.196**	0.240	0.011	-0.002
	6 h	**0.231**	0.191	0.009	0.247	0.234	0.010	0.017
	12 h	**0.268**	0.173	0.008	0.280	0.220	0.010	0.012
	15 h	**0.254**	0.138	0.006	0.275	0.185	0.008	0.021
LW_IN	3 h	**0.199**	0.172	0.008	0.204	0.193	0.009	0.006
	6 h	0.289	0.206	0.009	**0.286**	0.221	0.010	-0.003
	12 h	**0.337**	0.211	0.009	0.346	0.234	0.010	0.009
	15 h	**0.339**	0.183	0.008	0.343	0.212	0.009	0.004
VPD	3 h	**0.064**	0.065	0.003	0.097	0.101	0.005	0.033
	6 h	**0.097**	0.092	0.004	0.154	0.152	0.007	0.057
	12 h	**0.163**	0.135	0.006	0.235	0.210	0.009	0.072
	15 h	**0.171**	0.148	0.007	0.257	0.230	0.010	0.087
WS	3 h	**0.300**	0.186	0.008	0.300	0.185	0.008	0.001
	6 h	**0.395**	0.234	0.010	0.396	0.238	0.011	0.001
	12 h	**0.479**	0.250	0.011	0.499	0.264	0.012	0.020
	15 h	**0.470**	0.218	0.010	0.481	0.235	0.011	0.011
PA	3 h	**0.024**	0.016	0.001	0.028	0.017	0.001	0.004
	6 h	**0.037**	0.025	0.001	0.042	0.031	0.001	0.005
	12 h	**0.051**	0.025	0.001	0.055	0.026	0.001	0.004
	15 h	**0.055**	0.038	0.002	0.058	0.042	0.002	0.003
P	3 h	0.338	0.666	0.030	**0.323**	0.525	0.023	-0.015
	6 h	**0.381**	0.805	0.036	0.405	0.923	0.041	0.024
	12 h	**0.429**	0.572	0.026	0.456	0.680	0.030	0.027
	15 h	**0.485**	1.049	0.047	0.508	1.213	0.054	0.023
SWC	3 h	**0.013**	0.019	0.001	0.014	0.024	0.001	0.001
	6 h	**0.017**	0.026	0.001	0.021	0.033	0.001	0.004
	12 h	**0.030**	0.049	0.002	0.033	0.049	0.002	0.004
	15 h	**0.032**	0.045	0.002	0.040	0.055	0.002	0.008
TS	3 h	**0.028**	0.025	0.001	0.033	0.029	0.001	0.004
	6 h	**0.046**	0.048	0.002	0.061	0.058	0.003	0.015
	12 h	**0.088**	0.091	0.004	0.114	0.118	0.005	0.026
	15 h	**0.102**	0.104	0.005	0.132	0.130	0.006	0.030

Table A.5: Comparison between KF models fine-tuned to each variable (*KF-⟨var⟩-Sin-6_336*) and generic model trained for gaps in any variable (*KF-Gen-Sin-6_336*). The table displays the mean, the standard deviation (std) and the standard error (se) of the *Root Mean Square Error* (RMSE). In addition the difference (diff.) between the mean of two models is shown. The best method for each gap length is highlighted in bold. For each combination of gap length and variable, 500 artificial gaps were created.

	type	Generic			Fine-tuned one var			
	Stand. RMSE	mean	std	se	mean	std	se	diff.
Variable	Gap							
TA	6 h	0.092	0.056	0.002	**0.055**	0.033	0.001	-0.037
	12 h	0.118	0.074	0.003	**0.077**	0.052	0.002	-0.041
	1 day (24 h)	0.147	0.077	0.003	**0.101**	0.059	0.003	-0.046
	1 week (168 h)	0.174	0.064	0.003	**0.129**	0.048	0.002	-0.045
SW_IN	6 h	0.256	0.165	0.007	**0.209**	0.167	0.007	-0.047
	12 h	0.315	0.165	0.007	**0.258**	0.164	0.007	-0.057
	1 day (24 h)	0.339	0.129	0.006	**0.281**	0.132	0.006	-0.059
	1 week (168 h)	0.358	0.103	0.005	**0.298**	0.119	0.005	-0.060
LW_IN	6 h	0.299	0.201	0.009	**0.257**	0.200	0.009	-0.043
	12 h	0.368	0.197	0.009	**0.326**	0.209	0.009	-0.041
	1 day (24 h)	0.400	0.176	0.008	**0.359**	0.186	0.008	-0.042
	1 week (168 h)	0.466	0.141	0.006	**0.406**	0.165	0.007	-0.060
VPD	6 h	0.138	0.105	0.005	**0.097**	0.089	0.004	-0.041
	12 h	0.227	0.184	0.008	**0.169**	0.165	0.007	-0.058
	1 day (24 h)	0.273	0.207	0.009	**0.202**	0.160	0.007	-0.070
	1 week (168 h)	0.325	0.154	0.007	**0.264**	0.153	0.007	-0.061
WS	6 h	0.572	0.523	0.023	**0.365**	0.201	0.009	-0.206
	12 h	0.660	0.409	0.018	**0.482**	0.265	0.012	-0.178
	1 day (24 h)	0.676	0.374	0.017	**0.511**	0.270	0.012	-0.166
	1 week (168 h)	0.782	0.265	0.012	**0.569**	0.190	0.008	-0.212
PA	6 h	0.099	0.066	0.003	**0.054**	0.032	0.001	-0.045
	12 h	0.116	0.061	0.003	**0.060**	0.031	0.001	-0.056
	1 day (24 h)	0.137	0.076	0.003	**0.072**	0.069	0.003	-0.065
	1 week (168 h)	0.159	0.068	0.003	**0.081**	0.054	0.002	-0.078
P	6 h	**0.542**	0.991	0.044	**0.542**	0.991	0.044	0.000
	12 h	**0.627**	1.035	0.046	**0.627**	1.035	0.046	0.000
	1 day (24 h)	**0.628**	0.575	0.026	**0.628**	0.575	0.026	0.000
	1 week (168 h)	**0.881**	0.573	0.026	**0.881**	0.573	0.026	0.000
SWC	6 h	0.152	0.107	0.005	**0.051**	0.034	0.002	-0.102
	12 h	0.245	0.154	0.007	**0.067**	0.043	0.002	-0.177
	1 day (24 h)	0.393	0.223	0.010	**0.079**	0.053	0.002	-0.314
	1 week (168 h)	0.719	0.314	0.014	**0.169**	0.104	0.005	-0.550
TS	6 h	0.168	0.115	0.005	**0.065**	0.068	0.003	-0.103
	12 h	0.226	0.135	0.006	**0.088**	0.087	0.004	-0.138
	1 day (24 h)	0.286	0.152	0.007	**0.129**	0.128	0.006	-0.157
	1 week (168 h)	0.362	0.176	0.008	**0.317**	0.218	0.010	-0.046

Table A.6: The table displays the KF training difficulties. For a gap in one variable, three models are compared: a model trained with gaps in multiple variables ("Multi vars", *KF-Gen-Multi-6_336*), a model trained with gaps of only one variable ("Only one var", *KF-Gen-Sin-6_336*) and one model trained with gaps in multiple variables but initialized with random parameters ("Random params", *KF-Gen-Multi-6_336-Rand*). The models are expected to have comparable performances. The table displays the mean, the standard deviation (std) of the *Root Mean Square Error* (RMSE). The best method for each gap length is highlighted in bold. For each combination of gap length and variable, 500 artificial gaps were created.

	Train	Multi vars		Only one var		Random params	
	Stand. RMSE	mean	std	mean	std	mean	std
Variable	Gap [h]						
TA	3 h	**0.029**	0.024	0.066	0.050	0.060	0.045
	6 h	**0.044**	0.033	0.093	0.062	0.080	0.055
	12 h	**0.073**	0.050	0.133	0.092	0.117	0.085
	15 h	**0.077**	0.051	0.131	0.072	0.111	0.068
SW_IN	3 h	**0.178**	0.176	0.202	0.178	0.259	0.220
	6 h	**0.210**	0.170	0.249	0.176	0.283	0.209
	12 h	**0.269**	0.172	0.308	0.167	0.333	0.193
	15 h	**0.270**	0.166	0.315	0.164	0.334	0.188
LW_IN	3 h	**0.200**	0.169	0.215	0.177	0.298	0.203
	6 h	**0.270**	0.216	0.297	0.229	0.342	0.229
	12 h	**0.332**	0.224	0.377	0.241	0.398	0.259
	15 h	**0.357**	0.208	0.390	0.208	0.398	0.225
VPD	3 h	**0.064**	0.060	0.097	0.079	0.150	0.128
	6 h	**0.098**	0.096	0.148	0.113	0.193	0.174
	12 h	**0.168**	0.156	0.223	0.161	0.240	0.192
	15 h	**0.186**	0.159	0.241	0.168	0.254	0.210
WS	3 h	**0.305**	0.183	0.471	0.460	0.447	0.300
	6 h	**0.372**	0.210	0.576	0.466	0.482	0.309
	12 h	**0.429**	0.195	0.649	0.481	0.523	0.270
	15 h	**0.479**	0.230	0.691	0.457	0.540	0.293
PA	3 h	**0.025**	0.019	0.073	0.062	0.067	0.057
	6 h	**0.035**	0.021	0.095	0.061	0.098	0.077
	12 h	**0.051**	0.045	0.117	0.073	0.118	0.089
	15 h	**0.059**	0.040	0.128	0.080	0.115	0.076
P	3 h	**0.348**	0.680	0.411	1.510	0.371	0.713
	6 h	**0.400**	0.618	0.487	0.718	0.425	0.704
	12 h	**0.511**	0.851	0.706	1.150	0.535	0.981
	15 h	**0.453**	0.598	0.618	0.756	0.470	0.698
SWC	3 h	**0.012**	0.029	0.095	0.076	0.106	0.075
	6 h	**0.018**	0.031	0.151	0.108	0.153	0.107
	12 h	**0.027**	0.037	0.245	0.160	0.246	0.215
	15 h	**0.033**	0.041	0.282	0.228	0.278	0.208
TS	3 h	**0.024**	0.018	0.102	0.082	0.047	0.037
	6 h	**0.048**	0.048	0.164	0.106	0.076	0.074
	12 h	**0.083**	0.086	0.231	0.158	0.122	0.134
	15 h	**0.117**	0.129	0.271	0.191	0.154	0.161

Table A.7: Standard deviation of the meteorological variables for the entire Hainich FLUXNET 2015 dataset (σ_Y).

Variable	std	unit
TA	7.925	C
SW_IN	204.003	W/m^2
LW_IN	41.956	hPa
VPD	4.368	hPa
WS	1.625	mm
PA	0.855	m/s
P	0.280	W/m^2
SWC	8.913	%
TS	5.659	C

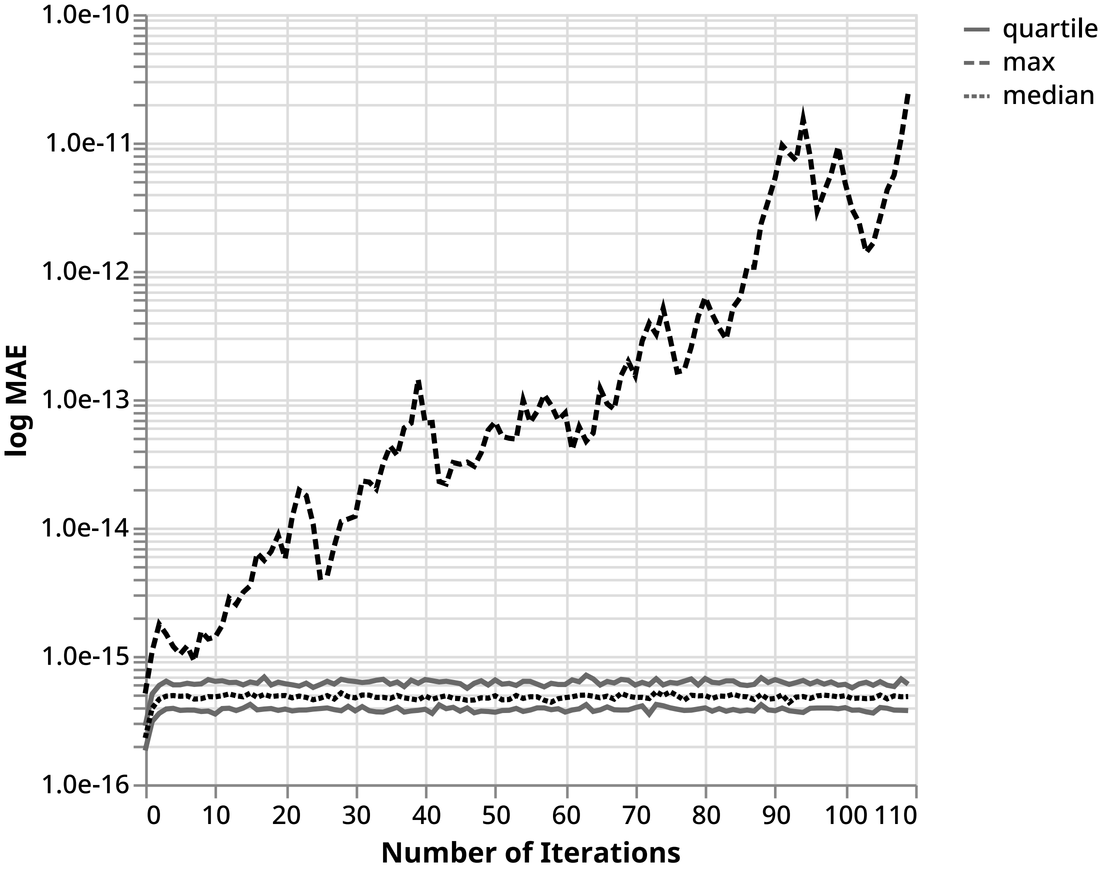


Figure A.6: Numerical stability comparison between Standard KF implementation and Square Root KF. For 100 times, the filters were initialized with a local trend model (drawn from a uniform distribution range 0-1) and then 110 observations processed using both models. At each time step, the Mean Absolute Error (MAE) was calculated between the state covariance from the Standard KF and the Square Root KF. The plot shows the median (black dotted line), 1st and 3rd quartile (gray lines) and the maximum of the MAE across the 100 samples (black dashed line).

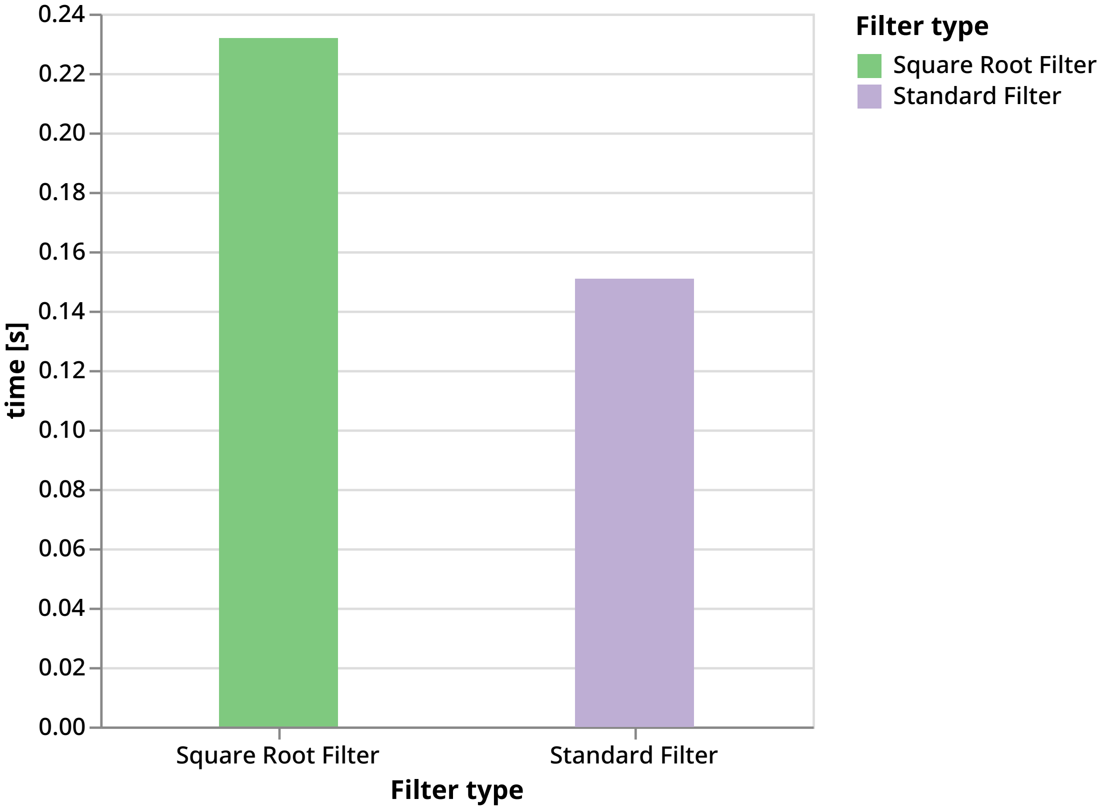


Figure A.7: Performance comparison between Standard KF and Square Root KF. The bar height shows the mean execution time for 100 samples with the following settings: number of observations: 100, dimension observations: 9, dimension state: 18, dimension control: 14, batch size: 20. The models were initialized with local trend parameters. The data was randomly generated by sampling a uniform distribution between 0 and 1, without any missing data.

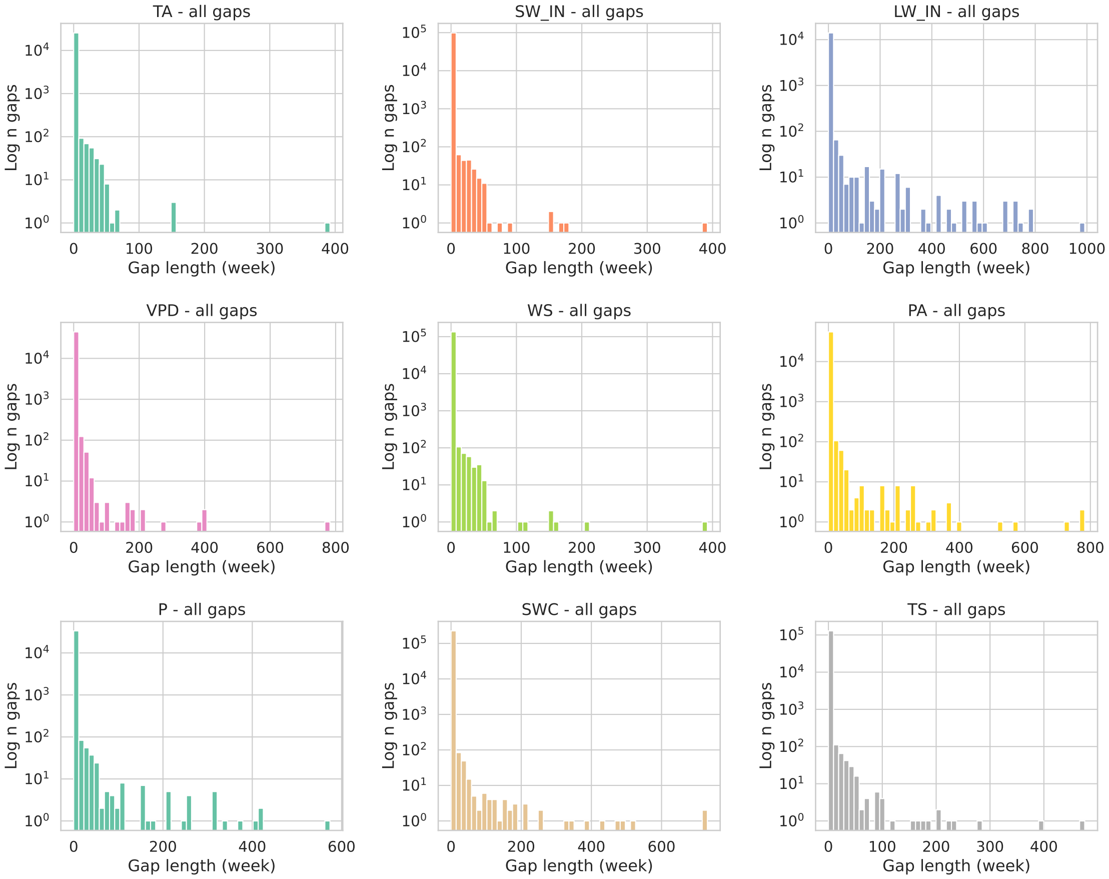


Figure A.8: Distribution of the gap lengths for all sites in the FLUXNET 2015 dataset for meteorological variables. The y axis displays the logarithm of the number of gaps. The length of the gap is the number of records where the QC flag is different than 0.

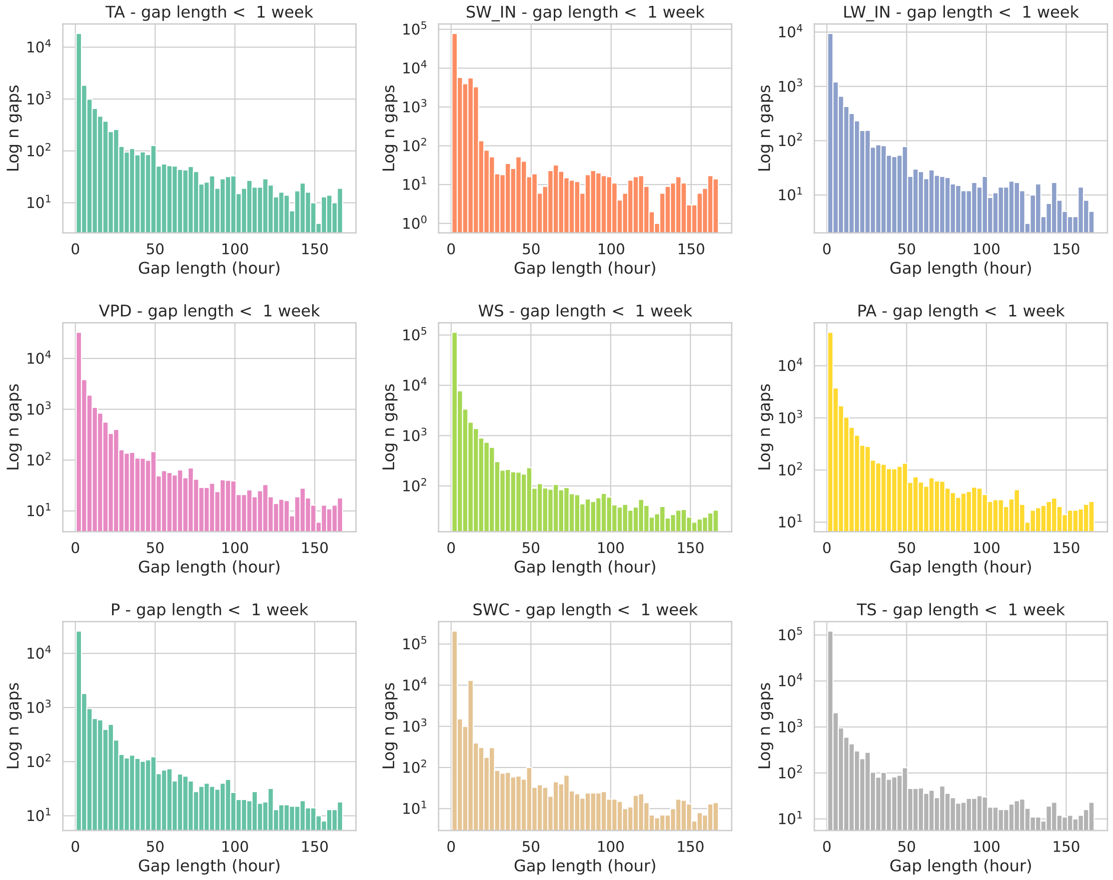


Figure A.9: Distribution of the gap lengths for gaps shorter than a week, for all sites in the FLUXNET 2015 dataset for meteorological variables. The y axis displays the logarithm of the number of gaps. The length of the gap is the number of records where the QC flag is different than 0.

B Derivations

$$
\begin{aligned}
p(y_t^g \mid Y^{ng}) &= \int p(y_t^g \mid x_t) p(x_t \mid Y^{ng}) dx_t \\
&= \int \mathcal{N}\left(y_t^g; MHx_t + Mb, MRM^\top\right) \mathcal{N}\left(x_t^s; m_t^s, P_t^s\right) dx_t \\
&= \mathcal{N}\left(y_t^{ng}; MHm_t^s + Mb, MRM^\top + MHP_t^s H^\top M^\top\right)
\end{aligned} \tag{B.1}
$$

$$
\begin{aligned}
MM^\top &= \begin{bmatrix} R^{1/2} & HP^- \\ 0 & (P^-)^{1/2} \end{bmatrix} \begin{bmatrix} R^{\top/2} & 0 \\ (P^-)^{\top/2} H^\top & (P^-)^{\top/2} \end{bmatrix} = \\
&= \begin{bmatrix} R^{1/2} R^{\top/2} + H(P^-)^{1/2}(P^-)^{\top/2} H^\top & H(P^-)^{1/2}(P^-)^{\top/2} \\ (P^-)^{\top/2}(P^-)^{1/2} H^\top & (P^-)^{1/2}(P^-)^{\top/2} \end{bmatrix} = \\
&= \begin{bmatrix} S & HP^- \\ (P^-)^\top H^\top & P^- \end{bmatrix} \\
VV^\top &= \begin{bmatrix} S^{1/2} & 0 \\ \bar{K} & P^{1/2} \end{bmatrix} \begin{bmatrix} S^{\top/2} & \bar{K}^\top \\ 0 & P^{\top/2} \end{bmatrix} = \begin{bmatrix} S^{1/2} S^{\top/2} & S^{1/2} \bar{K}^\top \\ \bar{K} S^{\top/2} & \bar{K}\bar{K}^\top + P^{1/2} P^{\top/2} \end{bmatrix} \\
&= \begin{bmatrix} S & S^{1/2} S^{\top/2} K^\top \\ K S^{1/2} S^{\top/2} & K S^{1/2} S^{\top/2} K^\top + P \end{bmatrix} \\
&= \begin{bmatrix} S & HP^- \\ P^- H^\top & KHP + P \end{bmatrix}
\end{aligned} \tag{B.2}
$$

All blocks of $VV^\top$ are directly equal to $MM^\top$, but the bottom left one, which is equal due to the measurement update for the covariance (Equation 4).

$$
\begin{aligned}
\text{RMSE}_{\text{stand}} &= \sqrt{\frac{\sum_i^n (y_i^z - \hat{y}_i^z)^2}{n}} \\
&= \sqrt{\frac{1}{n} \sum_i^n \left(\frac{y_i - \mu_Y}{\sigma_Y} - \frac{\hat{y}_i - \mu_Y}{\sigma_Y} \right)^2} \\
&= \sqrt{\frac{1}{n} \sum_i^n \left(\frac{y_i - \hat{y}_i}{\sigma_Y} \right)^2} \\
&= \frac{1}{\sigma_Y} \sqrt{\frac{\sum_i^n (y_i - \hat{y}_i)^2}{n}} \\
&= \frac{\text{RMSE}}{\sigma_Y}
\end{aligned} \tag{B.3}
$$

According to the Examination Regulations, I hereby confirm, that I have written the present thesis independently and without making use of any other sources and tools than those that are indicated. I assure that the written version of this thesis corresponds to the digital version.