S1 Supplementary Figures and Tables

Table S1: Predictor variables for GPP, TER, and NEE used by FLUXCOM RS and RS+METEO. List of acronyms: Enhanced Vegetation Index (EVI), fraction of absorbed photosynthetically active radiation (fAPAR), leaf area index (LAI), daytime land surface temperature (LST_{Day}) and nighttime land surface temperature (LST_{Night}), middle infrared reflectance (band 7; MIR), Normalized Difference Vegetation Index (NDVI), Normalized Difference Water Index (NDWI), plant functional type (PFT), incoming global radiation (Rg), air temperature (Tair), top of atmosphere potential radiation (Rpot), Water Availability Index lower (WAI_L). MSC denotes the mean seasonal cycle of a variable. AMP denotes the amplitude of the mean seasonal cycle, MIN the minimum of the mean seasonal cycle. See Tramontana et al. (2016) for further details.

	RS	RS+METEO
Spatial predictors	PFT AMP(EVI) AMP(MIR) AMP(LST_{Day})	PFT AMP(NDVI) AMP(band 4 reflectance) MIN(NDWI) AMP(WAI_L)
Spatial, seasonal predictors	MSC(LAI)	MSC(LST_{Night}) MSC(EVI*R_{pot}) MSC(fAPAR*LST_{Day})
Spatial, seasonal, interannual predictors	LST_{Day} LST_{Night} NDVI*Rg NDWI	WAI_L Tair MSC(NDVI)*Rg

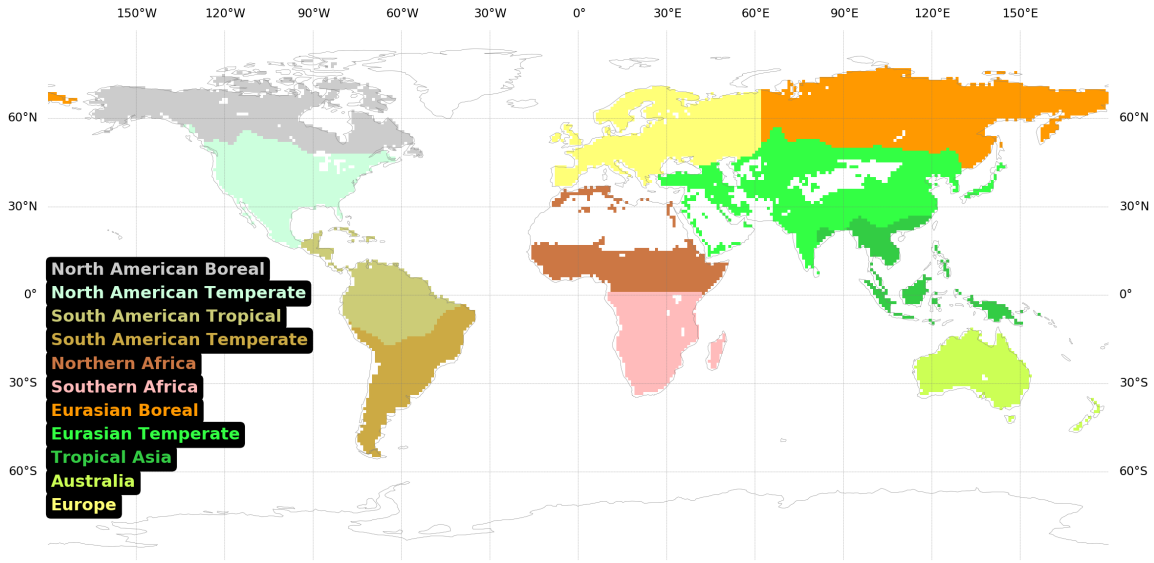

Figure S1: Map of TRANSCOM regions used for comparison with atmospheric inversions.

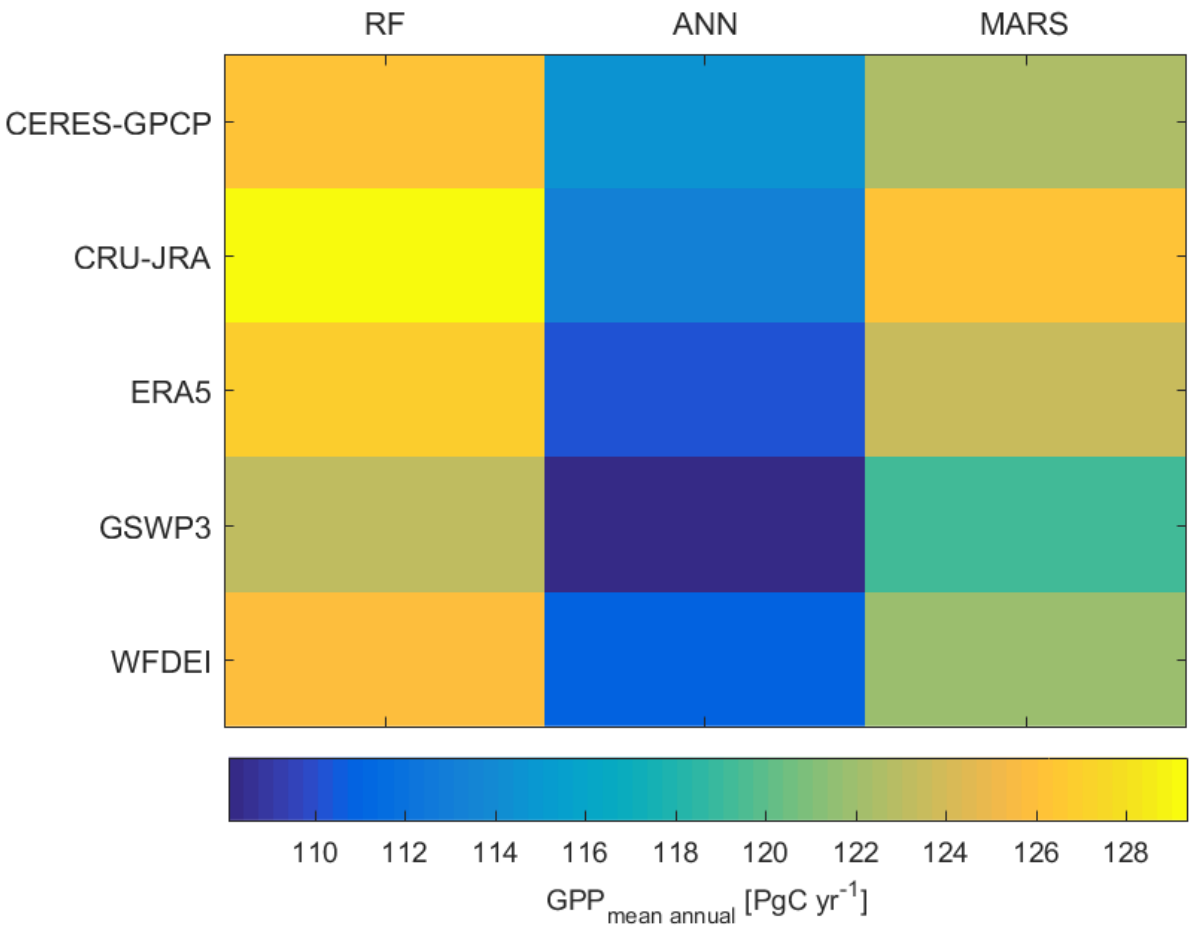

Figure S2: Global GPP of FLUXCOM-RS+METEO by machine learning method and meteorological forcing data.

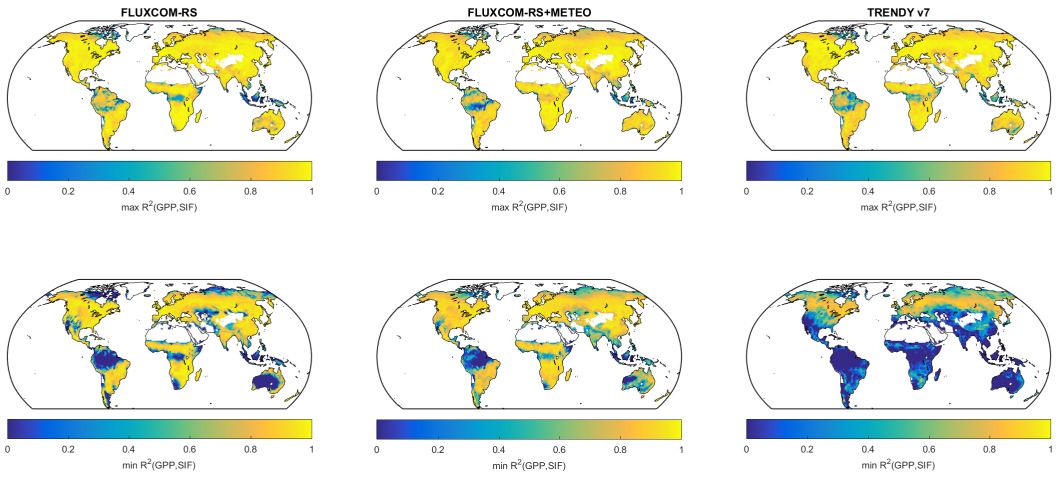

Figure S3: Maximum and minimum R^2(GPP, SIF) of FLUXCOM and TRENDY ensemble members.

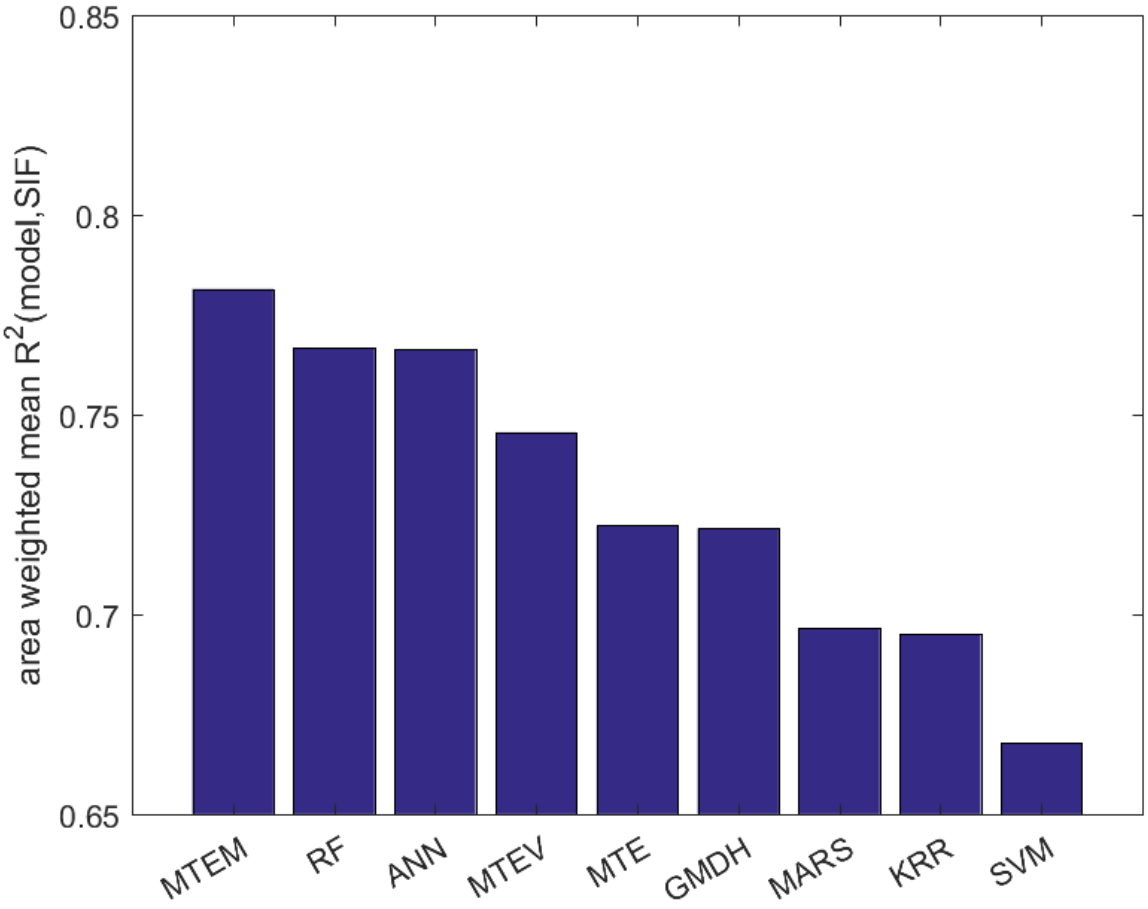

Figure S4: Mean R^2(GPP, SIF) of different machine learning methods of FLUXCOM-RS. List of Acronyms: Model Tree Ensemble Martin (MTEM), Random Forests (RF), Artificial Neural Network (ANN), Model Tree Ensemble Viterbo (MTEV), Group method of data handling neural network (GMDH), Multivariate Adaptive Regression Splines (MARS), Kernel Ridge Regression (KRR), Support Vector Machine (SVM).

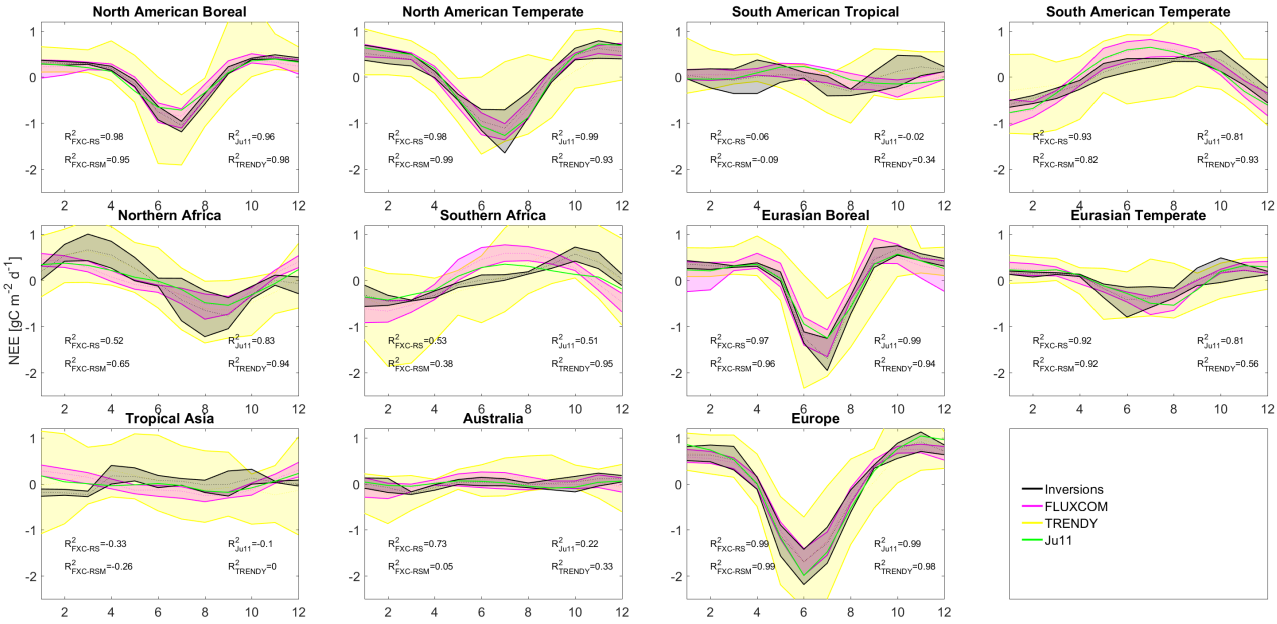

Figure S5: Same as Figure 7 in main text but based on NEE for FLUXCOM and TRENDY, and an approximation of inversion NEE by accounting for fire emissions.

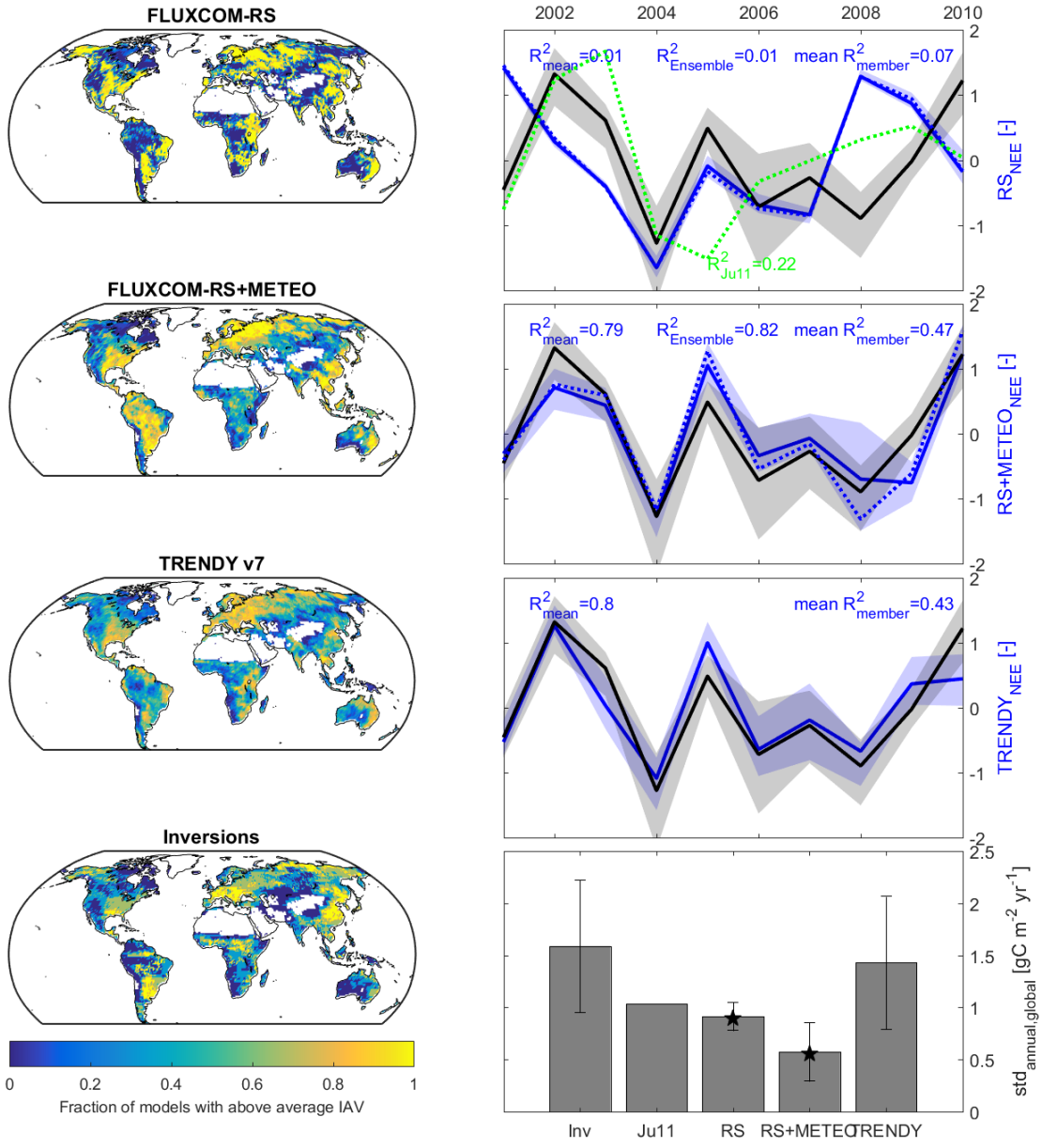

Figure S6: Same as Figure 8 in main text but based on NEE for FLUXCOM and TRENDY, and an approximation of inversion NEE by accounting for fire emissions.

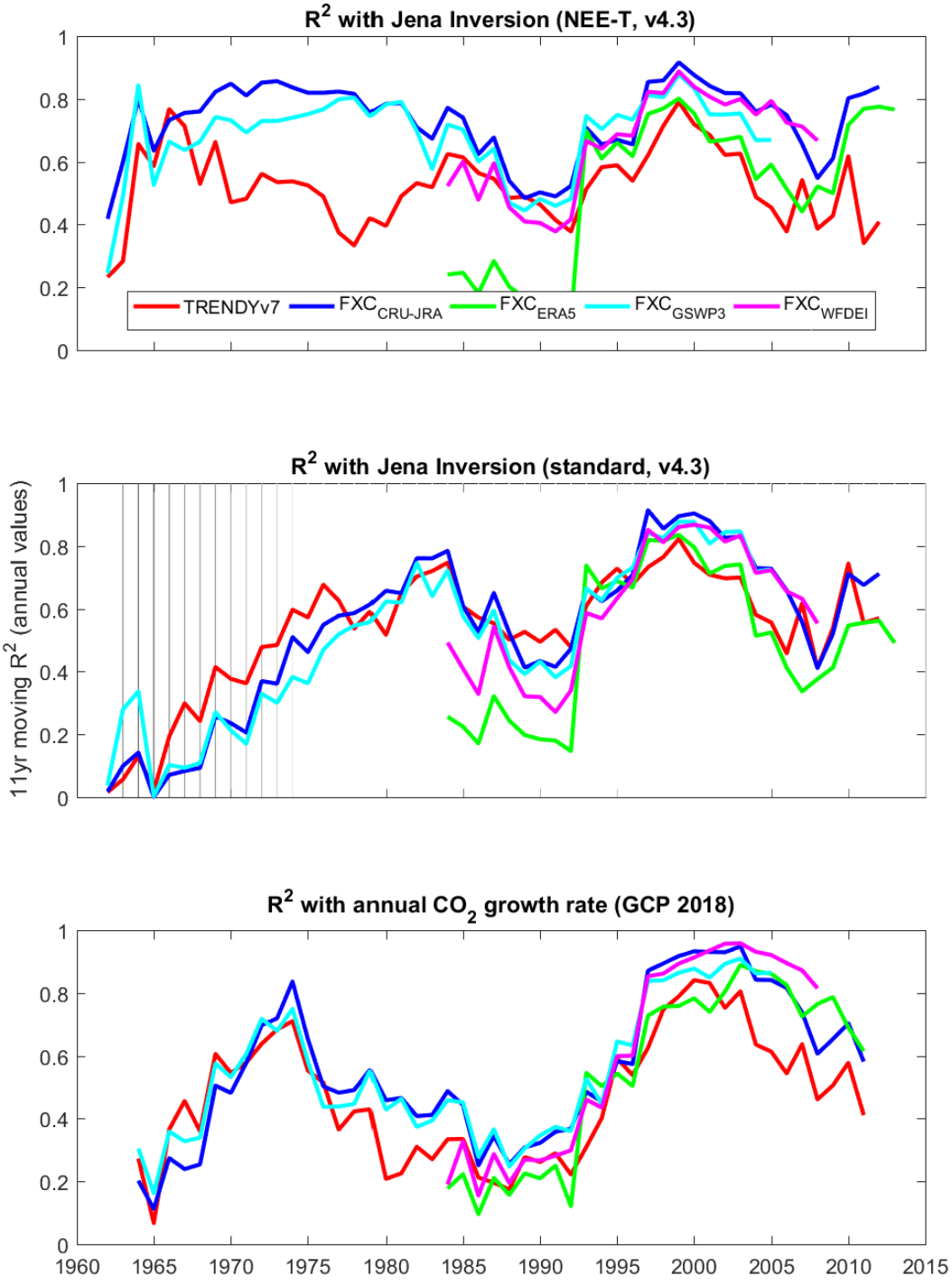

Figure S7: Moving window R^2 (11yrs, centered) of global NEE from FLUXCOM-RS+METEO with different meteorological forcings and TRENDY with two long-term inversions and CO_2 growth rate. Vertical gray stripes in the middle plot indicate where the correlation with inversions is likely deteriorated due to atmospheric station data-gaps (darker gray refers to more gaps).

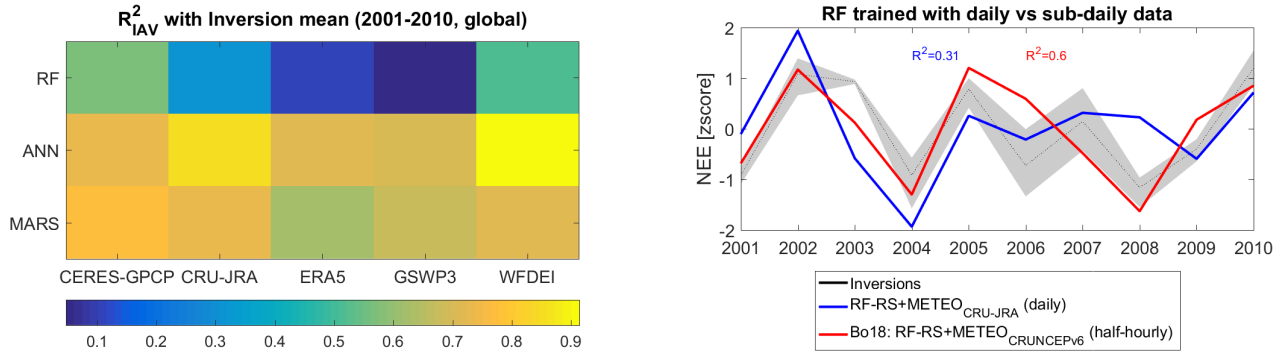

Figure S8: R^2 of global NEE from FLUXCOM-RS+METEO ensemble members with inversion mean for the period 2001-2010 (left) and comparison with estimates from Bodesheim et al. (2018) with inversions (right).

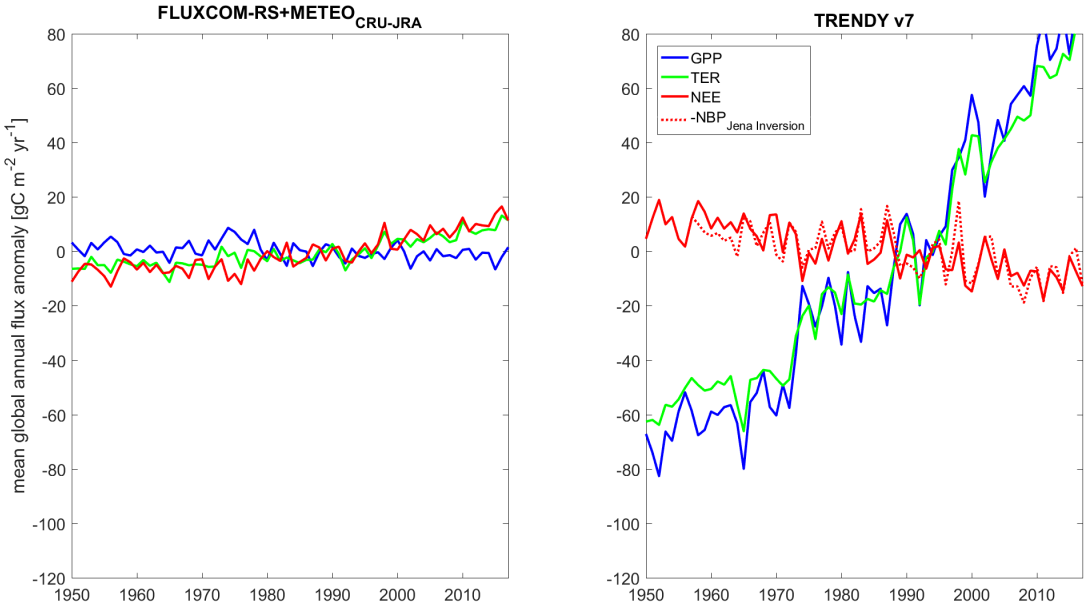

Figure S9: Annual anomalies of NEE, GPP, and TER from FLUXCOM-RS+METEO and TRENDY for 1950-2017. The Jena atmospheric inversion product is CarboScope s57Xoc_v4.3.

S2 Extrapolation and extrapolation severity index

S2.1 Principle and motivation

Extrapolation is the process of estimating a variable beyond the observed range, in other words making predictions beyond conditions found in the training data set. Quantifying (or summarizing) the degree of extrapolation in a single index is relevant to interpret and infer the robustness of machine learning predictions. Thus, extrapolation detection complements other measures of uncertainty quantification. However, detecting and quantifying extrapolation is challenging and there is currently no universally accepted approach in machine learning. Here we present the development of an Extrapolation Index (EI) and the Extrapolation Severity Index (ESI). The EI of a prediction is the expected relative error given the distance of the predictors to the nearest points in the training data set. The ESI provides an estimate of the expected absolute error for a given distance to training data in predictor space by taking the expected absolute error of a learned function for a prediction into account.

S2.2 Extrapolation Index and Extrapolation Severity Index

Let us fix notation first. We are given a training dataset $\mathcal{D} = \{(u_n, v_n)|n = 1, \ldots, N\}$, and a test dataset (for which we assess extrapolation) $\mathcal{T} = \{(x_n, y_n)|n = 1, \ldots, M\}$ where u and x are the predictors, and v and y are the target variable. Subscripts identify the sample and superscripts identify the predictors $u_n = [u_n^1, \ldots, u_n^d] \in \mathbb{R}^d$. Using $\mathcal{D}$, we learn a function f able to predict over new, unseen, test points $x_i \in \mathbb{R}^d$ and obtain estimates $\hat{y}_i \in \mathbb{R}$ of the true value $y_i \in \mathbb{R}$. For f the K nearest neighbors (KNN) of the test point x_i in the training dataset $\mathcal{D}$ are found and the mean target variable values v of the K nearest neighbors (KNN) is taken to obtain $\hat{y}_i$: $\hat{y}_i = \bar{v} = \frac{1}{K}\sum_{k=1}^{K} v_k$.

The Extrapolation Index (EI, unitless) for the test point $x_i \in \mathbb{R}^d$ is the average distance D to all its K nearest neighbors in the training dataset $[u_1|\cdots|u_K] \in \mathbb{R}^{d\times K}$, multiplied with ε, the relative expected prediction error change with distance of the learned function f:

$$EI_i = \varepsilon \times \frac{1}{K}\sum_{k=1}^{K} D_{i,k}. \tag{1}$$

We use the L_1-distance for measuring the distance D between two vectors x_i and u_k

$$D_{i,k} = \sum_{p=1}^{d} w_p |x_i^p - u_k^p|, \tag{2}$$

which is a weighted sum of absolute differences of the predictors d widely used in geostatistics and ecology. The weights w account for different relevance of predictor variables, which are optimized (see below) but could also be defined by the user from previous analyses.

We define ε as the mean change of the error δ of f with increasing distance D estimated for all N samples in the training dataset $\mathcal{D}$, normalized by the mean error of f:

$$\varepsilon = \frac{\sum_{n=1}^{N} \frac{\partial \delta_n}{\partial D_n}}{\sum_{n=1}^{N} \delta_n}, \tag{3}$$

where (1) δ_n is the absolute error of the predicted target variable by f, and (2) $\frac{\partial \delta_n}{\partial D_n}$ is estimated for each training sample n by calculating vectors δ_n and D_n by moving from smallest to largest distances, i.e. successively excluding nearest neighbors. Note that ε is a convenient scale factor of D to make EI interpretable and comparable between different target variables. This way EI becomes the expected relative prediction error increase with respect to increasing distance from training data in the predictor space. For example, an EI value of 0.2 refers to an expected 20% error increase due to the distance to the nearest neighbors in the training data.

The Extrapolation Severity Index (ESI, unit of target variable) scales the previous EI for x_i with an estimate of its expected error δ_i to obtain an expected absolute error due to distance to nearest training data:

$$ESI_i = \delta_i \times EI_i, \tag{4}$$

δ_i is estimated given x_i by a function learned on δ_n of a cross-validation. This function to estimate δ_i will not be perfect but provides some guidance on how the expected error changes in predictors space such as how the error scales with the magnitude of the expected target variable. In essence, ESI increases the expected error δ_i by EI which reflects the distance to nearest points to the training data.

S2.3 Application to GPP from FLUXCOM-RS

The above described principles of EI and ESI were applied to the target variable GPP from FLUXCOM-RS and considering the respective predictors set (see Section 4.1.2 and Figure 11 of main text). The parameters w and K were optimized using a Bayesian optimisation approach (function bayesopt in MATLAB) by minimizing $\sum_{n=1}^{N} \delta_n$ in a leave-one-site-out cross-validation (i.e. omitting data from the same flux tower site under test), where w was bounded between zero and one. For the categorical variable (PFT) we used a one-hot encoding strategy (transforming the variable in a series of binary variables, one for each category, but considering the same weight w for all binary variables of the same categorical variable). In the estimation of $\frac{\partial \delta_n}{\partial D_n}$ (also based on leave-one-site-out-cross-validation) the δ_n vector was smoothed by a LOWESS filter, and the mean derivative of the smoothed δ_n with respect to D_n was numerically approximated using the method of finite differences. Because w and thus also ε were sensitive to the training data set, the optimisation and estimation of ε was repeated five times based on subsampling the training data set and EI was computed as the median value of the 5 ensemble members. We used Random Forests to estimate δ_i which was trained on the absolute error of GPP predictions from the ensemble median of machine learning methods considered in FLUXCOM-RS. The absolute error was obtained from the cross-validation analysis of Tramontana et al. (2016).

S2.4 Advantages and limitations

The main strengths of the proposed indices are that they are (1) conceptually simple, interpretable, and quantitative, (2) that the considered predictor sets are not arbitrary but tied to a target variable and taking individual predictor importance for the target variable into account, (3) that parameters can be optimized in an objective way without manual or subjective intervention. There are also several limitations that need to be considered for interpreting EI and ESI: (1) The actual magnitudes of error increase due to extrapolation predicted by EI and ESI need to be interpreted cautiously because (a) they are based on a KNN model whose predictions are inferior to other machine learning models, (b) because the true prediction error with increasing distance ($\frac{\partial \delta_n}{\partial D_n}$) can show widely varying shapes and slopes depending on the sample and positioning in predictor space while we use a mean (ε), and (c) because we have an imperfect model for estimating δ_i. Thus primarily the patterns of the indices should be interpreted while comparing the magnitudes among different versions (e.g. target variables or predictor sets) can also be instructive. The above mentioned limitations are basically related to the problem that (a) the true extrapolation error for a test point is by definition unknown, and (b) extrapolation behavior varies by machine learning method and associated hyper-parameters. For the latter, we were interested in metrics that are largely based on the data and not very sensitive to machine learning method choice. We used KNN to estimate ε and Random Forests for δ_i because both do not explicitly fit functions but base their predictions on proximity in predictor space to the training data. Thus, we stay close to the data and avoid the usage of implicitly parametrized functions learned by machine learning algorithms like neural networks or kernel methods that, in the case of extrapolation, are very likely very model specific (because not constrained by the training data). (2) We assume sufficiency, i.e. that we have access to all relevant predictors. If important predictors are missing (or are severely undersampled), extrapolation

with respect to such cases cannot be detected. While the predictor sets used here for GPP were identified based on objective predictor selection the initial full set of candidate predictors may not be complete or the training data are too scarce to extract some relevant features. (3) The method is at present computationally expensive for large data sets as for each test point x_i, the distances to the K nearest neighbors in the training data need to be computed. This can be alleviated in the future by emulation, i.e. training a machine learning algorithm on EI estimated by our method over a smartly sampled set of test points.