Bachelor's Thesis

Analysis of Spatial Patterns in STARS-Generated Data using Climate Networks and κ-Statistics

Author:
Malte Viefhues
born 31.08.1986 in Nordhorn

Supervisors:
Prof. Dr. Jürgen Kurths
Dr. Norbert Marwan

Submitted to the

Department of Physics
Faculty of Mathematics and Natural Sciences
Humboldt-Universität zu Berlin

in partial fulfillment of the requirements for the degree of

Bachelor of Science in Physics.

Date of submission:

18 April 2016

Contents

1. Introduction **1**

2. Climate model and data **4**
2.1. Statistical Analogue Resampling Scheme 4
2.1.1. Motivation and mechanism 4
2.1.2. Limitations 5
2.2. Climate data 6
2.2.1. Observation and simulation data 6
2.2.2. Restructuring and preprocessing 8

3. Methodology **12**
3.1. Networks and climate 12
3.1.1. Basic network theory 12
3.1.2. Climate networks 13
3.2. Network measures 17
3.2.1. Node-splitting invariant degree 17
3.2.2. Spatially corrected strength 18
3.2.3. Spatially corrected directionality 19
3.3. κ-statistics for spatial agreement 20
3.3.1. Categorization of network measures 20
3.3.2. Cohen's original κ-statistic 21
3.3.3. Pontius' κ_{loc} for categorical maps 22
3.4. Significance estimation 24
3.4.1. Surrogate time series 24
3.4.2. Sensitivity to random noise 26

4. Analysis **29**
4.1. Comparison of STARS and basis data 30
4.1.1. Analysis by chunk time period 30
4.1.2. The effect of link density 32
4.1.3. Analysis by chunk mean temperature 34
4.2. Intercomparison of chunks 36
4.2.1. Analysis by chunk time period 36
4.2.2. Analysis by chunk mean temperature 40

5. Conclusion **42**

Appendices **49**

A. Climate model and data **50**
A.1. STARS model 50
A.2. Parameters and forcing used to generate S 54

B. Methodology **56**
B.1. Mathematical notes 56
B.2. Categorization of network measures 57
B.3. κ-statistics 58
B.4. Surrogates 64
B.5. Noisy time series 69

C. Analysis **71**
C.1. Comparison of STARS and basis data 71
C.2. Comparison matrices 72
C.3. Network measure distributions 79
C.4. Linear regressions 80

D. Other **83**
D.1. Acknowledgments 83
D.2. Statement of authorship 83

List of Figures

2.1. Meteorological stations used as nodes. 7
2.2. Pages2K Europe temperature reconstruction. 9
2.3. Illustration of phase-averaging. 10
2.4. Illustration of eventization. 11

3.1. Distribution of mean daily temperature values. 14
3.2. Distribution of total daily precipitation. 16
3.3. Categorized map of node strength. 23
3.4. Propagation of noise through the methodological tool chain. 27
3.5. STARS performance compared to surrogates and noisy time series. . . 27

4.1. Visual comparison of high and low spatial agreement. 29
4.2. Degree. Development of κ_l over time. 30
4.3. Strength. Development of κ_l over time. 30
4.4. Directionality. Development of κ_l over time. 31
4.5. Influence of link density on spatial agreement. 33
4.6. Distributions of similarity measures. 34
4.7. Degree. Development of κ_l with chunk mean temperature. 35
4.8. Strength. Development of κ_l with chunk mean temperature. 35
4.9. Degree. $\kappa_{l,m}$ between all chunks, ordered by time. 37
4.10. Mean temperature. $\kappa_{l,m}$ between all chunks, ordered by time. 39
4.11. Degree. $\kappa_{l,m}$ between all chunks, ordered by temperature. 40
4.12. Precipitation. $\kappa_{l,m}$ between all chunks, ordered by time. 41

A.1. Illustration of the STARS mechanism. 52
A.2. Example STARS time series and comparison with basis data. 53
A.3. Example frequency distribution in STARS and basis data. 54
A.4. Forcing of STARS during the hindcast. 54

B.1. Distribution of degree scores before and after categorization. 58
B.2. Illustration of network measure categorization. 58
B.3. Comparsion of z_l and $z(C_l, B)$ for mean temperature. 65
B.4. Comparison of z_l and $z(C_l, B)$ for precipitation. 66
B.5. Example distribution of $\kappa(C_l^s, B)$ for mean temperature. 67
B.6. Example distribution of $\kappa(C_l^s, B)$ for precipitation. 67
B.7. Illustration of surrogate shuffle mechanism. 68
B.8. Degree and Strength. STARS vs. surrogates and noisy time series. . . 69
B.9. Directionality. STARS vs. surrogates and noisy time series. 70

C.1. Directionality. Development of κ_l with chunk mean temperature. . . . 71
C.2. Comparison matrices for precipitation, ordered by time. 72
C.3. Comparison matrices for temperature, ordered by time. 73
C.4. Comparison matrices for temperature, ordered by temperature. 74
C.5. Comparison matrices for precipitation by link density, ordered by time. 75

C.6. Comparison matrices for precipitation by link density, ordered by temp. 75
C.7. Comparison matrices for min. temp. by link density, ordered by time. 76
C.8. Comparison matrices for min. temp. by link density, ordered by temp. 76
C.9. Comparison matrices for mean temp. by link density, ordered by time. 77
C.10. Comparison matrices for mean temp. by link density, ordered by temp. 77
C.11. Comparison matrices for max. temp by link density, ordered by time. 78
C.12. Comparison matrices for max. temp. by link density, ordered by temp. 78
C.13. Example distributions of network measure scores of an ESN. 79

List of Tables

2.1. Example data from the basis dataset. 7
2.2. Overview of simulation data chunks. 9

3.1. Overview of climate variables, network types, and network measures. 20
3.2. Generic contingency table. 22

4.1. Mean and std. dev. of κ_l per network measure. 32
4.2. Mean and std. dev. of κ_l and $\kappa_{l,m}$ by various criteria. 38

A.1. STARS parameter file used in hindcast. 55

B.1. Expected proportion of chance agreement in categorical maps. 59
B.2. Overview of different κ-types. 59

C.1. Mean and std. dev. of κ_l per network measure, ordered by temp. . . 80
C.2. Mean and std. dev. of κ_l for degree per link density, ordered by temp. 81
C.3. Mean and std. dev. of κ_l for degree per link density, ordered by time. 82

1. Introduction

With the ever-increasing availability of high-performance computing, much of modern climate science relies on computer-generated model data. These simulations are used to gain insights into possible developments of the climate system, examine scenarios based on different parametrization and expected trends, and derive predictions or recommendations for policy makers. Naturally, simulating the unknown is prone to errors and uncertainties stemming from incomplete knowledge of the system state and its key physical processes, as is testified by the limited reliability of weather forecasts beyond even a handful of days. Climate models do not try to accurately simulate weather at the same resolution of the forecasts we see on TV, however. Instead, they are designed to take a broader perspective, incorporating key physical characteristics of the climate system in question that are expected to show their influences on much larger time scales. These long-term trends are often approximately constant at the scale of regular weather forecasting, and can only be assessed with specifically adapted models.

Every climate simulation, whether generated by a dynamical or a statistical model, is based on assumptions and estimations. They lie at the heart of the design of a model's fundamental operating principles, and they influence how the model is parametrized when applied to a particular problem, emphasizing certain physical processes over others. In short, given the complexity of the climate system, no one model fits all. It is all the more important to analyze climate model outputs independently, to assess the strengths and weaknesses of every mechanism, define sensible parameter ranges, and offer guidelines on when these models can be deployed and how they can be interpreted correctly. Some common approaches for analysis and validation include the comparison with output from other climate models or from the same model using different parameters, plausibility checks through statistical analysis of the simulation data, or cross-validation with observation data if observation period and simulation period overlap.

In this study, a regional statistical climate model called STARS (see [1] and [2]) is examined using a combination of these strategies. The STARS model mechanism resamples time series data from an observation period to generate data for a simulation period via date-to-date mapping. One feature of STARS is that it performs its calculations only on a single (or a set of very few) time series, and then applies the date-to-date mapping derived at that location to all time series in the data set. In other words, STARS resamples synchronously for all locations, operating on *slices* of data that cover the whole domain. This has important implications, one of which we aim to analyze in this study is the model's limited spatial variability. Since all time series are shuffled synchronously, the mechanism can at most maintain the spatial variability of the input data, and might never increase the variability. We will investigate how this issue impacts on STARS output when the model is applied to a relatively large territory.

We analyze a STARS hindcast that was carried out by Jan Volkholz at the Potsdam Institute of Climate Impact Research (PIK), simulating the climate between the years 1500 and 1999 at 2331 distinct geographic locations located in modern-day Germany. This is a novel application of STARS, which is routinely used only for forward-facing projections. Although the state of the climate system is unknown for the simulation period, there have been efforts – notably the Pages2K project [3] – to reconstruct average annual temperatures at the continental scale for the past 2000 years, mainly from tree-ring data and documentary records. Jan Volkholz used these reconstructed temperatures to derive the forcing parameters for STARS. In this sense, the forcing is conditioned on 'real' data covering the hindcast period and not, as is the usual case in STARS applications, on the projections of other models.

Over the past 500 years, several large-scale climatically relevant trends occurred in Europe. Two examples are the Little Ice Age, and the warming associated with industrialization [4], [5]. These phenomena are visible in the Pages2K data, e.g. resulting in a very cool period between 1580 and 1620 and a very warm period between 1970 and 2000, respectively [3]. They are of course expected to have influenced the climate in Germany and to have had varying impacts over time on the climate of different subregions. The Ruhr region, for example, was among the first and most heavily industrialized areas in our domain. We therefore expect some connection between the spatial patterns in climate data covering this period, and those long-term climate trends.

Limited interannual variability is a problem of many statistical models, and in the case of STARS has been discussed in the literature as well [6], [7]. We will focus on spatial variability, however: developed for a small regional scale, STARS lacks a mechanism driving spatially differentiated developments and there is an argument to be made that STARS further restricts spatial variability by resampling synchronously at all geographic locations while using only the data of one, or very few, locations in its physically-motivated calculations. However, it is still unclear where exactly the STARS approach ceases to be useful because it can no longer provide locally differentiated insights, or simply generates unphysical data.

Following the approach pioneered by Feldhoff et al. [8] and further developed by Lange et al. [6], we use complex networks for climate model evaluation. We divide our dataset covering 500 years into smaller chunks covering periods of 50 years. On each of these chunks, we use Pearson's correlation coefficient r and a statistical measure called event synchronization e to investigate the temporal relationship of climate variables between geographic locations. The climate variables we consider are the daily mean and extreme temperatures, and the total daily precipitation. To reveal the overall spatial structure of these relationships in a given chunk, we use r and e to construct climate networks and explore them using network measures specifically designed for spatially embedded problems. This process results in network measure scores for each geographic location, encoding information about the spatio-temporal structure contained in that particular chunk of our simulation data. In order to investigate the development over time, and to detect any changes that might indicate a presence of long-term climate trends in the data, we compare the network scores of our chunks using κ-statistics. We use a version of κ specifically

adapted to the use for model evaluation with categorical maps that has been introduced by Hagen-Zanker [9].

The structure of our study is as follows: in chapter 2 we will describe the climate model STARS, give a very short overview of its resampling mechanism, and indicate some of the problematic features arising from it. The chapter continues with a brief overview of the simulation and basis datasets we used, and the preprocessing we performed on these data. It also explains how we restructured the simulation data into periods of 50 years.

We follow in chapter 3 with a description of our methodology that combines approaches from complex network theory and the statistical evaluation of categorical maps. We show in detail how spatially embedded networks are generated from the simulation and basis data, how these networks can be explored with the help of so-called network measures, and how the spatial structure exposed through these network measures can be analyzed using κ-statistics. This chapter concludes with some sensitivity tests of the methodology that will help with the interpretation of our results.

In the analysis chapter 4, we investigate how the spatial structures in the simulation data, as measured through our climate networks, have developed over the 500-year hindcast period. In addition, we examine a possible link between the average temperature present in the data and similarity as measured by κ-statistics.

Finally, we summarize our findings in chapter 5, draw some conclusions from our study, review some of our methodology, and provide a few ideas ideas for future studies.

2. Climate model and data

2.1. Statistical Analogue Resampling Scheme

The *Statistical Analogue Resampling Scheme* (STARS) for regional climate simulations was developed by Orlowsky et al. and is described in detail in [1] as well as [2], [10] and [11]. STARS has been used in numerous climatological studies and analyses, in regional research projects such as GLOWA-Elbe, and as a link in the model chain of the outreach portal Climate Impacts Online [12]–[19]. In the context of model evaluation, STARS-generated data have been used as well, often covering considerable geographical dimensions such as the whole South American continent [8].

2.1.1. Motivation and mechanism

STARS was designed to be as independent from the output of dynamical models as possible, a common source of uncertainty in other statistical models. *Statistical climate models* rely on stochastic relationships among observed data to generate simulated data. They are often more accurate than dynamic models, but rely on high-quality, high-resolution data that might only be available in some regions. In order to connect to the larger-scale climate system, statistical models are commonly parametrized with the output of dynamical models.

Dynamical climate models use differential equations to simulate the most relevant physical processes, for example atmospheric circulations. They can operate at the global scale but are computationally expensive and vulnerable to small variations in their parametrization. STARS aims to reduce this source of error by radically reducing the number of outside parameters it depends on and relying almost completely on the observation data used as its input.

A detailed overview of the mechanism is provided in App. A.1. In short, the model resamples observed time series data to assemble simulated time series, resulting in a date-to-date mapping between the observation and simulation periods. The scheme is forced by the two parameters a and b of a linear trend line

$$\omega = a \cdot t + b \tag{2.1}$$

describing the development of some climate variable ω, e.g. mean temperature, during the simulation period. STARS resamples the original observations in a way to closely match a and b, while respecting heuristic rules designed to ensure the physical properties of the system such as annual cycles or persistence. The necessary calculations are performed with a small number of representative time series only – in our case only one –, but the date-to-date mapping derived there is applied to all time series in the dataset.

Formally, the scheme incorporates a number of different approaches to statistical climate modeling:

1. The statistic sampling is forced by the parameters of a linear regression of some characteristic variable. In this regard, STARS employs a simple transfer function that uses the characteristic variable as predictor and everything else as response variables.

2. Similar to weather generator models, STARS uses heuristic rules to introduce physical criteria into a stochastic process of resampling observed data. For example, it restricts sampling of data by calendar date preventing January data appearing in July.

3. STARS follows the logic of a simple analog scheme. Let the system be in state

$$X(t_1) = \Big(x(t_1), y(t_1), z(t_1)\Big),$$

and let $y(t_1)$ be unknown. The idea is to search the dataset to find the state

$$X(t_2) = \Big(x(t_2), y(t_2), z(t_2)\Big)$$

most similar, i.e. *analog*, to $X(t_1)$ with regard to $x(t_1)$ and $y(t_1)$, and then set $y(t_1) \equiv y(t_2)$.

Importantly, STARS depends on one characteristic variable ω that 'must be sufficiently relevant for the dependent variables' [2]. In this sense, STARS is a downscaling mechanism, taking only the (aggregate) values of the predictor to generate high-resolution simulation data. In doing so, it largely removes the dependency from dynamical model output and the uncertainty of parametrization.

In our study, this predictor was derived from the reconstruction data of the Pages2K project, and STARS was used to generate a hindcast covering 1500 to 2000 by downscaling the time series provided by Pages2K.

2.1.2. Limitations

Data generated with the STARS mechanism have some limitations, some of which are common for all statistical models while others are due to its rather radical approach to parametrization. We will discuss three important problems.

First, the trend line has to capture the climatic variability of the whole simulated domain. This might be very difficult for larger areas that might span topologically very different and geographically far apart locales. The model also assumes that different climate variables essentially co-develop in time, and that the trend of the predictor is consequently correlated with the trend of the co-variables.

Second, the temporal variability of the observation data has to be high enough to allow the developments expected in the simulation period to be actually reflected in the simulation data. The model restricts interannual variability, for example, as discussed by Lange et al. in [6] and Feldhoff et al. in [8]. Closely tied to this is the criticism put forward by Wechsung and Wechsung in [7] that STARS invalidly

builds long-term climate gradients from short-term inter-annual gradients.

Third, STARS applies the same date-to-date mapping at all geographic locations in its domain, the observation data is processed in 'slices' covering the entire domain for one time step. The spatial variability of the simulation data is in this sense severely restricted, and there is no mechanism ensuring locally independent or even differentiated developments.

In this study, we will focus on the last problem. The simulation run analyzed covers the area of Germany over a period of 500 years, so is well outside the scale for which STARS was initially developed by Orlowsky, who worked on the Elbe basin subregion of Germany and with periods not exceeding 100 years.

We will use complex network theory and κ statistics to examine this large dataset in order to explore the plausibility of the simulation output, especially with regard to its spatial variability.

2.2. Climate data

We work with climatological station data, networks calculated from these data are thus spatial networks as described below in Sec. 3.1.1. In contrast to grid data, station data are provided at the specific geographic coordinates of meteorological stations and the underlying domain is usually not uniformly sampled. This is the case for our data as well, as different stations are taken to be representative of varying surface areas. The data is organized as one time series per station, providing daily values for a set of eleven climatologically relevant variables.

2.2.1. Observation and simulation data

We differentiate between the observational data, called the basis data or real data B, and the data generated by the STARS model from these observations, called the simulation data S.

Basis data

The raw observations at the core of our basis data B are provided by the German Meteorological Service (Deutscher Wetterdienst, DWD) as daily time series between 1951 and 2006 at 2342 climate and precipitation stations around Germany, Switzerland and Austria.

The dataset was tested by Österle et al. for physical consistency of extreme values and between different climate variables, and for spatial consistency between neighboring stations. Where necessary, they interpolated missing or inconsistent values to create a homogeneous dataset [20].

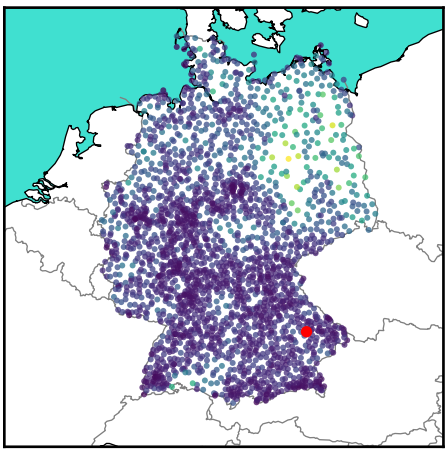


Figure 2.1.: The meteorological stations that are the source of our observation time series and constitute the geographically embedded nodes analyzed in this study. Note the irregular sampling density, illustrated by marker color, and artificial (political) borders. STARS resampling was performed at the highlighted station.

What we call our basis dataset B is a subset of these data, covering Germany only and consisting of $N = 2331$ daily time series of length 17897 (see Fig. 2.1).[1] Out of the eleven variables provided by the DWD, we use the minimum, mean and maximum daily temperatures T_{min}, T_{mean}, T_{max}, as well as total daily precipitation P. An example of the data provided, and its precision, is given in Tab. 2.1.

Table 2.1.: Example data from the basis dataset, station 17007 in Potsdam. The columns used in this study are `tmax`, `tmean` and `tmin`, each in °C, and `prec`, given in mm. The simulation data is structurally identical.

day	mo	year	tmax	tmean	tmin	prec	relh	pres	watv	sunh	clou	inso	wind
1	1	1951	-1.5	-3.8	-12.6	1.5	83.0	987.7	3.8	1.5	7.3	297.0	4.3
2	1	1951	-0.3	-2.5	-5.1	0.0	92.0	983.8	4.9	0.0	5.0	208.0	4.3
3	1	1951	1.2	-0.2	-5.0	0.0	85.0	991.2	5.2	0.0	7.7	141.0	2.9
4	1	1951	3.4	0.6	-1.0	0.0	82.0	1003.6	5.4	2.2	4.0	198.0	5.8
5	1	1951	4.1	2.4	-1.1	4.8	92.0	999.0	6.6	0.0	7.0	158.0	5.8

Simulation data

Our simulation dataset S is a hindcast for the period between 1500 and 2000 that includes the same stations, and thus covers the same domain, as our B dataset. It was generated with single-station STARS simulations by Jan Volkholz (Potsdam Institute for Climate Impact Research, Research Domain 2) using reconstructed temperature data from the Pages2K project to create trend lines, and a time series from B as input.

The Pages2K project is a collaborative effort to reconstruct continent-scale annually-resolved temperature and temperature variability values for the period between the years 1 and 2000 [3]. For Europe, the series combines eleven proxy datasets from

[1] For compatibility with prior work, we do not use the DWD data at non-German stations and some stations on the northern islands, and consider only data between 1 January 1951 and 31 December 1999.

annually-resolved documentary records and tree-ring analysis: the authors computed average summer temperatures for cells in a $5° \times 5°$ grid covering the land-only parts of Europe and aggregated them by area-weighted averaging for the whole domain. The result is a single annually-resolved time series of reconstructed temperatures for Europe.

In what can be interpreted as a downscaling of the Pages2K data, which offers one value per year covering the Pages2K Europe region ($13.0 \cdot 10^6$ km^2), Jan Volkholz used STARS to generate the S dataset at the resolution of B, providing $N = 2331$ values per day, covering Germany ($35.7 \cdot 10^3$ km^2). He proceeded in five-year increments, dividing the Pages2K time series into 100 periods and performing a linear regression according to Eq. (A.3) for each period (see Fig. A.4). Each set of parameters a and b was then used to force a 5-year STARS resampling of the T_{mean} time series at the DWD station in Straubing[2] (see Fig. 2.1). Finally, he concatenated these results to form full 500-year time series.[3] Any additional STARS parameters used by Jan Volkholz can be found in Tab. A.1.

An example of the simulation time series, and the comparison to the equivalent basis time series, can be found in Figs. A.2 and A.3.

2.2.2. Restructuring and preprocessing

Prior to the computation of our similarity measures, and thus the creation of climate networks, as will be described in the next section, the original datasets B and S were restructured to allow for our time-based analysis, and the data underwent some preprocessing.

Restructuring

The 500-year time series of STARS data were divided into periods of 50 years that we call *chunks*, for two main reasons: first, to examine a potential relationship between time and network spatial structure as measured by κ (see below). Second, for better comparability with the basis data time series, which roughly cover 50 years.

The simulation data is thus organized into ten chunks S_l, $l = 1, \ldots, 10$, covering the periods 1500-1550, 1550-1600, ..., 1950-1999. For each chunk, we have 2331 time series with a length of around 18170 daily values, depending on leap years. Networks created from simulation data, and measures calculated on them, are hence always associated with only one specific 50-year period, or one chunk, between 1500 and 1999.

[2] It is the author's guess that since the station in Straubing is close to the geographic center of the domain covered by Pages2K, it might be among those best represented by the Pages2K temperature time series.

[3] As can be seen in Fig. A.4, this approach has some obvious drawbacks. First, the trend lines do not connect, i.e. the overall line is not continuous, with rather significant jumps between the 5-year periods. Second, the regressions are carried out with only five data points, leading to less robust results. Third, S is not strictly speaking one unified dataset, and some of the STARS heuristics explained in App. A.1 are essentially made ineffective, namely the restriction of block re-use.

For convenience, we introduce a continuous numbering of STARS and basis chunks so that $C_l \equiv S_l$ for $l \in \{1, \ldots, 10\}$ and $C_{11} = B$. An overview of all chunks can be found in Tab. 2.2.

Table 2.2.: The chunks used in this study. The mean seasonal temperature per chunk $\langle T_l(\mathcal{S})\rangle$ is calculated according to Eq. (4.2) and given in °C. $\langle T_l(\text{all})\rangle$ is very close, but not identical, to the 50-year average that can be computed from the Pages2K data. The ranks are obtained by ordering the chunks according to the respective temperatures. Except for S_2, which ends on the 29 February, all chunks begin on 1 June and end on 28 February of the respective start and end years.

Chunk	S_1 C_1	S_2 C_2	S_3 C_3	S_4 C_4	S_5 C_5	S_6 C_6	S_7 C_7	S_8 C_8	S_9 C_9	S_{10} C_{10}	B C_B
Start	1500	1550	1600	1650	1700	1750	1800	1850	1900	1950	1951
End	1550	1600	1650	1700	1750	1800	1850	1900	1950	1999	1999
$\langle T_l(\text{all})\rangle$	8.01	7.81	7.62	7.75	7.80	8.04	7.78	7.99	8.17	8.33	8.22
Rank	7	5	1	2	4	8	3	6	9	11	10
$\langle T_l(\text{JJA})\rangle$	16.07	15.98	15.84	15.86	15.90	16.20	15.93	16.32	16.26	16.59	16.30
JJA rank	6	5	1	2	3	7	4	10	8	11	9
$\langle T_l(\text{DJF})\rangle$	-0.21	-0.35	-0.93	-0.50	-0.69	-0.08	-0.50	-0.30	0.01	0.25	0.23
DJF rank	7	5	1	4	2	9	3	6	8	11	10

The Pages2K data in Fig. 2.2 show that in Europe, the Little Ice Age was most pronounced during 1570 to 1630 (corresponding to our chunks C_2 and C_3), with 1570 to 1670 being an overall cold period (C_2, C_3, C_4). Higher temperatures that might be associated with the industrialization can be found between 1850 and 2000 (C_8, C_9, C_{10}), with the warmest period having occurred between 1970 to 2000 (C_{10}).

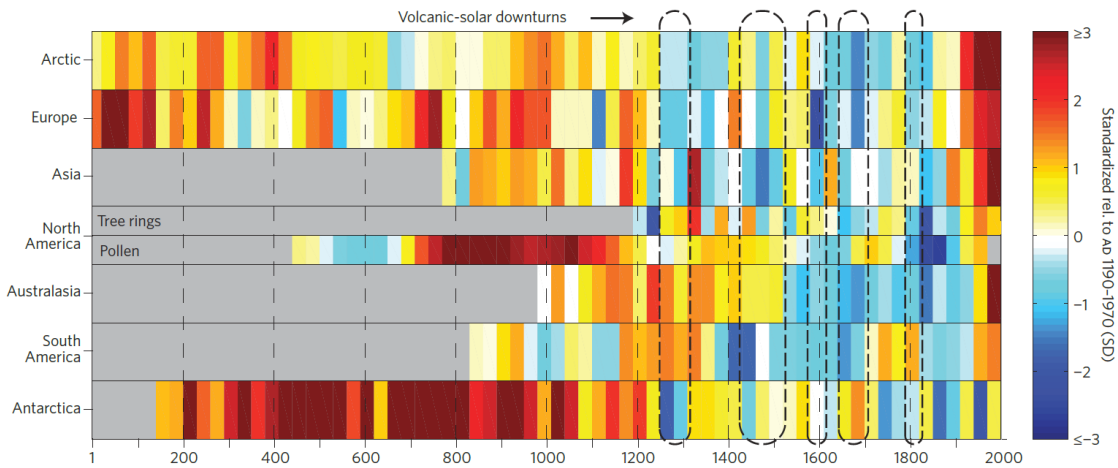


Figure 2.2.: Figure taken from [3]. It shows the continent-scale temperature Pages2K reconstructions as 30-year-mean standardized temperatures. The dashed boxes indicate pronounced volcanic and solar negative forcing. For more details see the original paper.

We separate observations according to their season $\mathcal{S}$ into *summer* and *winter* series. This is done routinely in climate research to suppress the effect of the annual cycle, for example on correlations. In order to do so, we extract those observations pertaining to the summer months June, July and August (JJA), or the winter months

December, January, February (DJF). Since all our similarity measures work at zero lag, we can concatenate these observations into a new time series without generating spurious additional correlations.

For example, the winter data of a given chunk contain observations from 1 June of the first year included until the 31 August of the last year, omitting all data not in DJF, for a typical length per time series of $n \approxeq 4500$, again depending on leap years.

Preprocessing

The original temperature time series $x_i(t)$ were transformed into so-called *climate anomaly series* $\tilde{x}_i(t)$ in a procedure called phase-averaging. Climate anomalies are obtained by subtracting the mean value of all data pertaining to the same day of the year - the long-term daily mean - from the original value. The mean of a phase-averaged time series is zero, and the climate anomalies have a roughly normal distribution, as shown in Fig. 2.3.

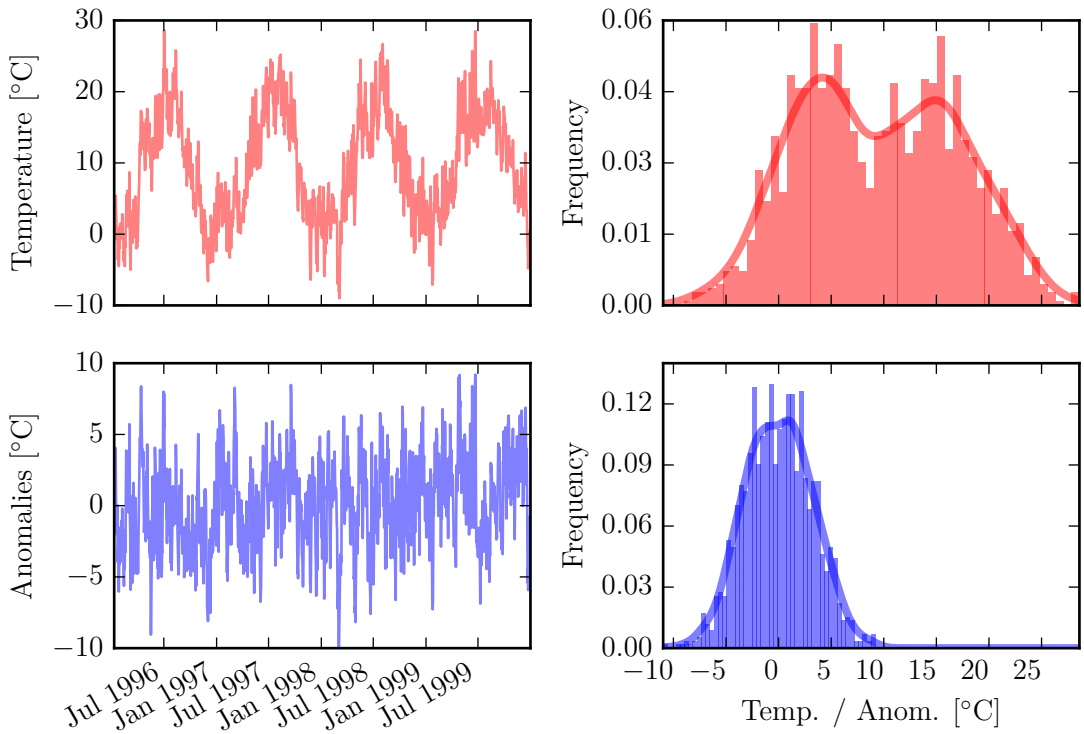


Figure 2.3.: Phase-averaging illustrated. We subtract the long-term daily mean. The lines in the histograms are kernel density estimates of the distributions (Gaussian kernel), illustrating the removal of the annual cycle (two peaks in the observation data).

The similarity measure we use for precipitation data works on binary *events*, as will be explained in Sec. 3.1.2. Hence, those data were not transformed into climate anomalies but converted directly to such *event time series*, denoted with $\hat{x}_i(t)$, by static thresholding. The threshold discerning events and non-events was chosen at $\theta = 10$mm, as illustrated in Fig. 2.4, in line with earlier work on German rainfall

data by Rheinwalt et al. [21]. We refrained from focusing on extreme precipitation, as is often done in studies covering South America such as [6], since their analysis carries additional difficulties and they are a less substantial feature of the climate system in Central Europe.

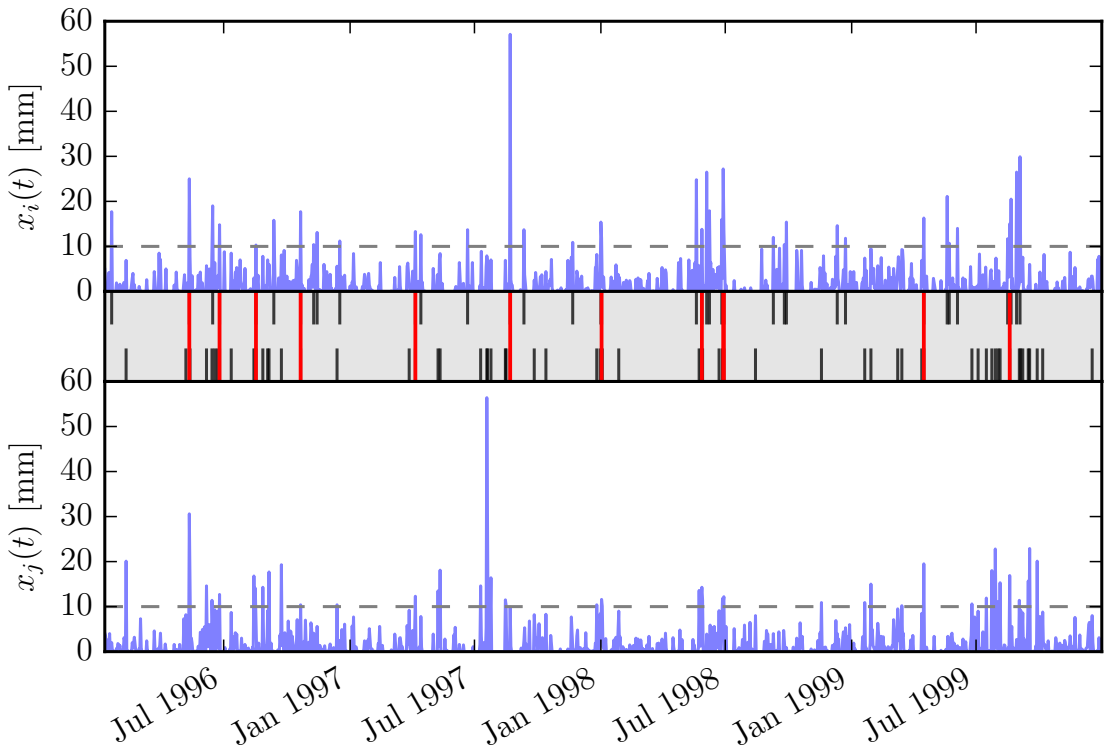


Figure 2.4.: Conversion of precipitation time series $x_i(t)$ and $x_j(t)$ to binary event series $\hat{x}_i(t)$ and $\hat{x}_j(t)$. The threshold $\theta = 10$mm is drawn as horizontal dashed lines. The shaded center plot illustrates the concept of event synchronization: black lines mark events in $\hat{x}_i(t)$ and $\hat{x}_j(t)$, respectively. Red lines indicate synchronized events.

For convenience, we will omit the hat and tilde notation in the following, always assuming that we work with the climate anomaly data and event time series, respectively, except where explicitly stated otherwise.

3. Methodology

In the early 2000s Tsonis and Roebber started to apply the mathematical tools of complex network theory to climate science [22]. The idea is to reveal the spatio-temporal dynamics of the climate system by turning the statistical relationships between the climate at different geographic points into links in a network, and performing analysis on these networks instead of the raw, often very large, datasets. This approach has been used by several studies in the past to analyze for example monsoon systems [23], [24] and El Niño [25].

Feldhoff et al. [8] pioneered the use of complex networks in the study of climate model performance over South America. Specifically, they compared the structures of simulation and reanalysis data, using STARS as one of their models under review, with the help of Hamming distance and root-mean-square error of clustering coefficients. In [6], Lange et al. continued in this vein and further developed the difference metric used to compare networks, again working inter alia with STARS data. In contrast to these studies, however, we rely on κ-statistics to compare networks, as will be explained further below.

3.1. Networks and climate

This section will briefly review some fundamental concepts from network theory, and how they are applied to climate data.

3.1.1. Basic network theory

Mathematically, a *network* is a finite set of nodes connected by links. Common examples are people linked by certain types of relationships (social networks) or neurons joined by synapses (neurological networks). The mathematical subfield concerned with the study of networks, also called graphs, is called graph theory.

Unweighted, undirected, simple networks

Consider an unweighted network (graph) $\mathcal{G}$ with a set of nodes $\mathcal{N} = \{1, 2, ..., N\}$ and a set of links $\mathcal{L} \subseteq \{(i, j) : i, j \in \mathcal{N}\}$ between them. The graph can be described mathematically via a binary $N \times N$-matrix, the so-called *adjacency matrix* $\mathcal{A}$:

$$\mathcal{A} = \begin{pmatrix} a_{00} & a_{01} & \cdots & a_{0N} \\ a_{10} & a_{11} & & \\ \vdots & & \ddots & \\ a_{N0} & & & a_{NN} \end{pmatrix} = (a_{ij})_{i,j \in \mathcal{N}}, \tag{3.1}$$

where

$$a_{ij} = \begin{cases} 1, & \text{if} \quad (i,j) \in \mathcal{L} \\ 0, & \text{else.} \end{cases}$$

If the network is *undirected*, then $a_{ij} = a_{ji}$, and $\mathcal{A}$ is symmetric. If a node cannot be linked to itself the network is *simple*, and the diagonal elements disappear: $a_{ii} = 0 \; \forall \; i$.

We call the proportion of ones to zero in $\mathcal{A}$ the network's *link density* ϱ,

$$\varrho = \frac{1}{N(N-1)} \sum_{i,j}^{N} a_{ij} \cdot 100 \; \%, \tag{3.2}$$

i.e. at a link density of 50% exactly half of the theoretically possible links in the networks are present.

Link-weighted networks

For many real-world applications, unweighted networks are insufficient, as they do not allow to further characterize the link between two nodes. This can be done with link-weighted networks which can be represented by a *weights matrix*

$$\mathcal{W} = (w_{ij})_{i,j \in \mathcal{N}}, \tag{3.3}$$

where each weight w_{ij} describes the relationship between a pair of nodes i and j. A higher value of w_{ij} can, for example, imply a closer relationship. In undirected networks, weights can often be chosen to satisfy $w_{ji} = w_{ij} \geq 0$ [26].

Spatial networks

In geographically (spatially) embedded networks, nodes are located in some Euclidean space S with a distance metric $\Delta : S \times S \longrightarrow \mathbb{R}^+$. An example are the nodes in our networks that represent meteorological stations (Fig. 2.1), described by their latitude and longitude values $(\varphi, \lambda)_i$. These allow the creation of a distance matrix containing the geodesic (great circle) distance Δ_{ij} between all nodes, and the definition of node weights v_i that are based on the area covered by each station (see below).

In real-world applications, spatial networks require additional attention since their topology is constrained, distance-related costs of links might occur [26], and the effects of (irregular) sampling and boundaries on network measures have to be considered, especially in networks covering a relatively small area such as the territory of Germany [21], [27].

3.1.2. Climate networks

Climate data is organized in time series $x_i(t) = \{x_i(t_1), x_i(t_2), \ldots, x_i(t_n)\}$ that contain climate variable observations at discrete points in time recorded at a particular location. The data structure thus implies a particular discretization of the underlying geographical and temporal space. We handle daily data, so $x_i(t_m)$, is the

observation on a particular day and $t_m \in \{t_1, \ldots, t_n\}$. In our study i denotes a meteorological station with the coordinates $(\varphi, \lambda)_i$, but gridded data is also common.

Climate networks connect the time series $x_i(t)$ and $x_j(t)$ (their nodes i and j) based on similarity measure scores between them (their links a_{ij} or w_{ij}). The calculation of a network thus collapses the temporal dimension while keeping the spatial structure of the data, leading to spatially embedded networks.

In our study, we use the correlation coefficient r to measure the similarity between temperature time series, and event synchronization e between precipitation time series. These measures are used as link weights w_{ij} in $\mathcal{W}$.

Pearson's correlation coefficient

Pearson's empirical correlation coefficient r_{ij} is a well-known measure of linear relationships between variables. Given two time series $x_i(t)$ and $x_j(t)$, it is defined as

$$r_{ij} = r_{ji} = \frac{\mathrm{cov}(x_i(t), x_j(t))}{\sigma_{x_i}\ \sigma_{x_j}} \tag{3.4}$$

where $\mathrm{cov}(x_i(t), x_j(t))$ is the sample covariance between $x_i(t)$ and $x_j(t)$, and $\sigma_{x_\bullet}$ is the sample standard deviation of $x_\bullet(t)$ (see App. B.1).

Pearson's r is normalized, so $-1 \leq r_{ij} \leq +1$. The sign of r_{ij} indicates a positive (negative) linear relationship between x_i and x_j, i.e. $x_i(t) = a \cdot x_j(t) + b$, with $a > 0$ if $r_{ij} > 0$ and $a < 0$ if $r_{ij} < 0$. A result of $r_{ij} = 0$ indicates that the two variables are uncorrelated. In our case, $x_i(t)$ and $x_j(t)$ are anomaly time series of a meteorological variable at two weather stations i and j.

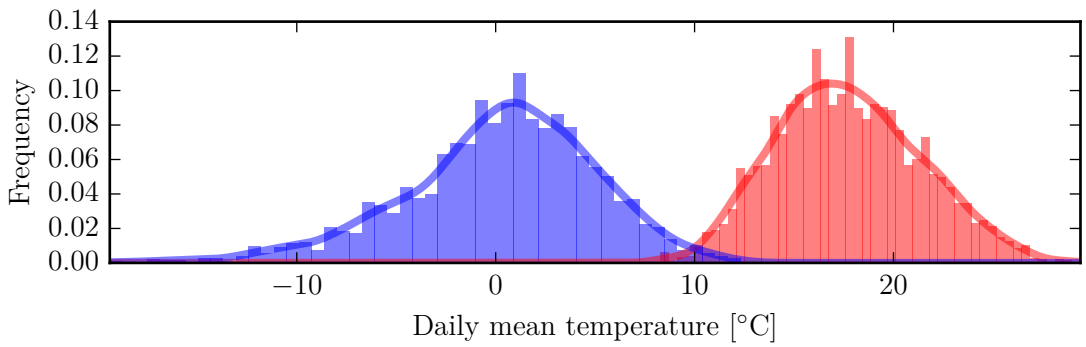


Figure 3.1.: Normalized histograms of mean daily temperature observations. The distributions are essentially normal. The lines represent kernel density estimations (Gaussian kernel) of both distributions. The data is from the STARS dataset between 1951-1999 for the station located in Potsdam, separated by summer (red) and winter (blue) months.

The correlation coefficient only quantifies linear relationships and works best on normally distributed variables without many outliers [28]. These requirements are met by temperature data (Fig. 3.1), but are problematic when it comes to precipitation (Fig. 3.2).

Event synchronization

Event synchronization is a non-linear measure that was first introduced by Quiroga et al. [29], and has been applied to STARS-generated data by Lange et al. [6]. It can be calculated between two time series with definable *events*. Each observation in an event series is either zero (no event), or one (event). In this study we restrict ourselves to a simple variant of event synchronization without lag.[1]

Let $x_i(t)$ and $x_j(t)$ denote two binary event series of length n, so that $x_\cdot(t_m) \in \{0, 1\}$ with $t_m = t_1, \ldots, t_n$. The number of events in $x_\cdot(t)$ is called their *event rate* and denoted as $c_\cdot$. Event synchronization counts the fraction of event pairs matching in time between the two series, as shown in Fig. 2.4. The *synchronization count* is the number of concurrent events in $x_i(t)$ and $x_j(t)$:

$$c_{ij} = c_{ji} = \sum_{t=t_0}^{t_n} x_i(t) \cdot x_j(t). \tag{3.5}$$

c_{ij} depends on c_i and c_j. More events increase both the maximum possible number of synchronizations, and the number of randomly occurring synchronizations. At the same time, the upper bound of c_{ij} is restricted by $\min(c_i, c_j)$. Following [21], we correct for this bias.

The expected number of chance synchronizations is estimated as the probability mean of the hypergeometric distribution $p(c)$[2]:

$$p(c) = \frac{\binom{c_i}{c}\binom{n - c_i}{c_j - c}}{\binom{n}{c_j}}. \tag{3.6}$$

We can then report the *event synchronization score* e_{ij}, calculated as the difference between observed and expected counts in units of the standard deviation of $p(c)$:

$$e_{ij} = e_{ji} = \frac{c_{ij} - \langle c \rangle_p}{\sigma_p}, \tag{3.7}$$

where $\langle c \rangle_p$ is the probability mean of $p(c)$, and σ_p its standard deviation (see App. B.1). The standardization of scores reduces the effect of very high synchronization scores between spatially close geographic points.

Event synchronization is a non-linear measure that works well with non-normally distributed variables. This enables the use with precipitation data that generally has a highly left-skewed non-Gaussian distribution (Fig. 3.2).

[1] Prior work with rainfall time series over the territory of Germany has indicated that the introduction of lag does not change the overall results drastically, given the limited geographic scope of the analysis.

[2] This models event synchronization as a drawing without replacement, with the events in both time series occurring at uniformly random time positions.

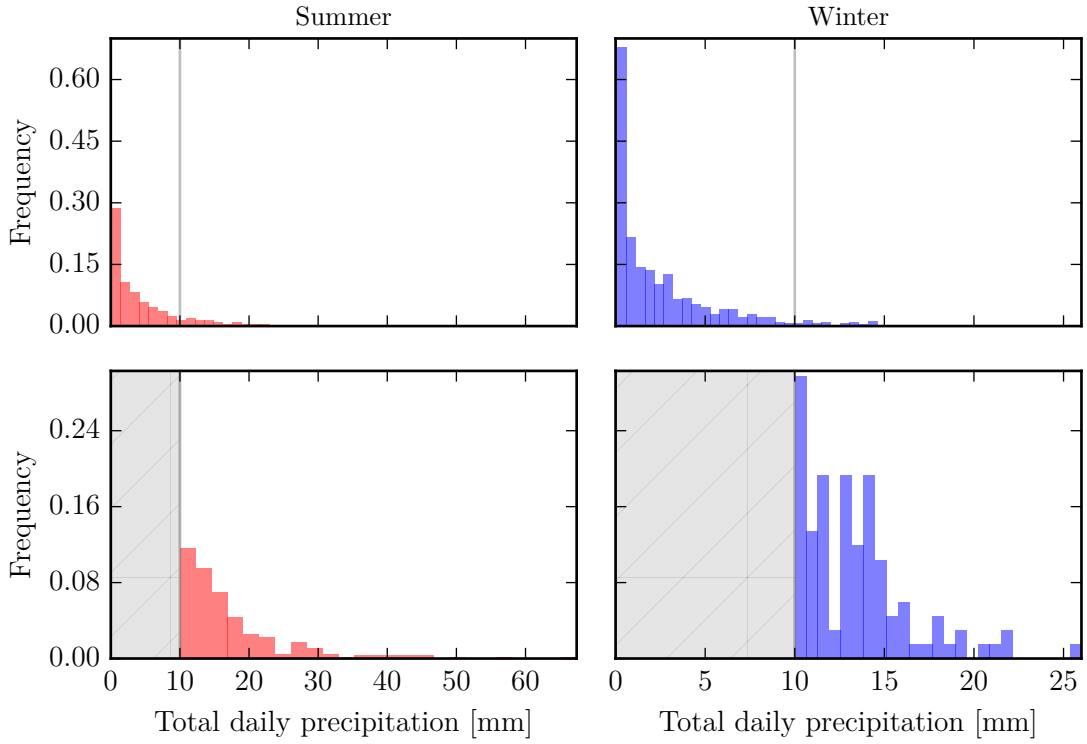


Figure 3.2.: Normalized histograms of total daily precipitation observations. The non-Gaussian distribution and presence of many extreme values can clearly be seen. The two upper plots include all non-zero observations, the lower plots include only observations above 10mm as used in ESN construction. The data is from the STARS dataset between 1951-1999 for the station located in Potsdam.

Link-weighted climate networks

Our two types of (static and spatially embedded) climate networks are constructed from temperature and precipitation data by calculating r_{ij} and e_{ij}, respectively, between all time series, and using them as link weights w_{ij} in the weights matrix $\mathcal{W}$. We call r and e the *similarity measure* of the network.

The use of r as weights in climate networks was pioneered by Tsonis and Roebber in [22], so we call the resulting (temperature-based) networks *Tsonis Climate Networks*, or TCN. Link weights in TCNs are representative of the linear correlation between nodes.

Networks based on e, on the other hand, are called *Event Synchronization Networks*, or ESN. We use them for precipitation data, and their link weights indicate how often precipitation occurs simultaneously at two nodes.

The raw time series data were subject to some preprocessing (see Sec. 2.2.2), namely transformations into climate anomaly series $\tilde{x}_i(t)$ for temperature data, and binary event series $\hat{x}_i(t)$ for precipitation data. Instead of using the raw values in $x_i(t)$, TCNs and ESNs were constructed from $\tilde{x}_i(t)$ and $\hat{x}_i(t)$, respectively.

Unweighted climate networks

Where calculations are performed on unweighted networks, for example during the computation of node-splitting invariant degree (see below), we derived the binary adjacency matrix $\mathcal{A}$ by thresholding from the raw similarity values in $\mathcal{W}$.

$$a_{ij} = \begin{cases} 1, & \text{if} \quad w_{ij} \geq \theta \\ 0, & \text{else,} \end{cases} \tag{3.8}$$

for some suitable threshold θ. In these unweighted climate networks a link is thus created if the similarity, for example the linear correlation, between two nodes is above a certain level.

The exact choice for θ is somewhat arbitrary, but network measure scores and link density are often correlated, so it is convenient to choose θ so that each network has a fixed link density, so $\theta = f(\mathcal{W}, \varrho)$ [30], [31]. Where applicable, we denote the link density of a network as a percentage value after the network type, e.g. we abbreviate a Tsonis climate network at 50% link density 'TCN50'.

3.2. Network measures

Networks can be studied with the help of 'importance measures' assigning one value per node, i.e. N values per network. Our analysis focuses on what we call a *network measure pattern*, the spatial distribution of importance across the embedded networks.

From the wide range of available network measures, three were selected that have been employed in climate network analysis in the past. Two are variants of the 'node degree' for spatially embedded and weighted networks, and one is a relatively novel measure introduced specifically to analyze precipitation patterns on climate networks [21], [27], [31]. The combinations of network types and measures used in this study are referenced in Tab. 3.1.

3.2.1. Node-splitting invariant degree

The *node splitting invariant degree* is an extension of the well-known degree measure. The degree of a node i is defined as

$$k_i = \sum_{j \in \mathcal{N}_i} a_{ij}, \tag{3.9}$$

where a_{ij} is an element of the network's binary adjacency matrix $\mathcal{A}$ and $\mathcal{N}_i$ denotes the network neighborhood of i, so $\mathcal{N}_i = \{j \in \mathcal{N} : a_{ij} = 1\}$ [26].

In real-world applications like climate networks, the discretization of the underlying continuous domain of interest is not straightforward. Weather stations are irregularly placed on the map and their measurements are taken to be representative for some area around them. If the sampling is sufficiently dense, adding a redundant station in the area already covered by another station should not yield any additional information, but will influence the absolute values of our degree measure.

Heitzig et al. propose a scheme called 'node splitting invariance' to counter such sampling effects. The revised version of degree in that scheme uses node weights v_i and is given by

$$k_i^* = \sum_{j \in \mathcal{N}_i^+} v_j, \tag{3.10}$$

where $\mathcal{N}_i^+ = \{j \in \mathcal{N} : a_{ij}^+ = 1\} = \mathcal{N}_i \cup \{i\}$ is the unpunctured neighborhood of node i, and $a_{ij}^+ = a_{ij} + \delta_{ij}$ is an element of the extended adjacency matrix $\mathcal{A}^+$ [27]. More information on node-splitting invariance can be found in App. B.1.

According to Heitzig et al., optimal weights should be 'inversely proportional to the sampling probabilities' of each node. The weight v_j of node j is thus, in this study, chosen to be the area covered by the weather station, as approximated by the Voronoi cell around it.[3]

3.2.2. Spatially corrected strength

Another approach to the problem of spatial effects in networks is used for the measure node strength. Strength is defined as

$$s_i = \sum_{j \in \mathcal{N}} w_{ij}, \tag{3.11}$$

with w_{ij} an element of the weights matrix [26, p. 199].

Rheinwalt et al. described the effect of artificial boundaries such as political borders on network measures computed on spatially embedded networks [21], [33]. They propose the use of a 'spatial surrogate network' to correct these effects. Following [21], the surrogate is a link-weighted network on the same set of nodes as the original (node count and spatial embedding), with the same dependency of link probability on the binned spatial length of links, i.e. the binned great circle distance Δ_{ij} between nodes.

The surrogate link weight w_{ij}^s between two nodes i and j is set to

$$w_{ij}^s = p(\Delta_{ij}) = \left[\sum_{k \in \mathcal{N}_i^{\Delta_{ij}}} 1 \right]^{-1} \cdot \sum_{k \in \mathcal{N}_i^{\Delta_{ij}}} w_{ik}, \tag{3.12}$$

where Δ_{ij} is the geodesic distance between nodes i and j, and Δ_{ij}^* denotes the geodesic distance binned using a granularity of $\varepsilon = 1500$m. $\mathcal{N}_i^{\Delta_{ij}} = \{k \in \mathcal{N} : \Delta_{ik}^* = \Delta_{ij}^*\}$ are those nodes that lie in the same distance bin from i as j. $p(\Delta_{ij})$ is then the weighted probability that a pair of nodes separated by a distance Δ_{ij} are linked in the original network [21].

[3] Taking our spatial networks as an example, the two-dimensional Voronoi cell of node i covers all points in our domain that are closer to $(\varphi, \lambda)_i$ than to any other node j, as measured by geodesic distance [32]. This is only a very rough approximation of real meteorological station coverage, as it does not account for local topographical features, for example mountains restricting the actual domain coverage of individual stations.

Network measures calculated on the surrogate can be interpreted as an estimate for spatial effects influencing the network measure in the original network, including both boundary effects and inhomogeneous sampling densities. Following Rheinwalt et al., we thus correct the strength by subtracting this boundary effect estimate from the values of the original network:

$$s_i^* = s_i - s_i^s \qquad \text{where} \qquad s_i^s = \sum_{j \in \mathcal{N}} w_{ij}^s, \tag{3.13}$$

with the superscript s denoting the values computed on the surrogate.

3.2.3. Spatially corrected directionality

The third network measure employed in this analysis is a spatially corrected version of the 'directionality' network measure introduced by Boers et al. [21], [24], [34]. It indicates the directions along which the most events occur synchronously, both in form of an orientation and a strength corresponding to the number of links pointing in that direction [34]. Since directionality is closely linked with event synchronization, we did not compute it on TCNs.

Directionality is calculated with the help of the distribution of synchronizations by direction $P_i(\varphi)$. Using bins to group similar angles together, it is given by

$$P_i(\varphi) = \sum_{j \in \mathcal{N}_i^\varphi} e_{ij}, \tag{3.14}$$

where φ is a fuzzy angle and $\mathcal{N}_i^\varphi = \{j \in \mathcal{N} : \Phi^*(i,j) = \varphi\}$ is the neighborhood of i of nodes that lie in the direction of φ. Here, $\Phi(i,j)$ denotes the geographic angle between nodes i and j, and $\Phi^*(i,j)$ its binned version using a granularity of $\varepsilon = 0.05°$. In our undirected networks we use $\Phi(i,j) \bmod \pi$ so that $\Phi(i,j) = \Phi(j,i) \in [0,\, \pi)$. e_{ij} are the synchronization scores between nodes (Eq. 3.7).

The directionality orientation φ_i and the directionality strength ρ_i of node i, are calculated directly from $P_i(\varphi)$:

$$\rho_i = \max_{\varphi \in [0,\, \pi)} P_i(\varphi) \tag{3.15}$$

$$\varphi_i = \arg\max_{\varphi \in [0,\, \pi)} P_i(\varphi). \tag{3.16}$$

Rheinwalt et al. point out that 'since $P_i(\varphi)$ is not a count of links, but a count of standardized synchronizations for a given fuzzy angle φ', it should be called a 'frequency distribution of synchronizations and not of links' [21].

Again, we use a spatially corrected version of directionality drawing a similar type of 'spatial surrogate network' as for strength (see Eq. (3.12)). The corrected measures are

$$\rho_i^* = \rho_i - c_i P_i^s(\varphi), \tag{3.17}$$

$$\varphi_i^* = \varphi_i - c_i P_i^s(\varphi), \tag{3.18}$$

with the superscript s denoting the values obtained from the surrogate. The constant c_i is defined as the quotient of the two average frequencies of synchronizations,

$$c_i = \frac{\langle P_i(\varphi) \rangle}{\langle P_i^s(\varphi) \rangle}, \tag{3.19}$$

and designed to ensure that the corrected directionality (ρ_i^*, φ_i^*) stays very close to the null vector if links are isotropic [21].

For convenience, we will omit the starred notation from now on, and assume our measures to be node-splitting invariant or spatially corrected, respectively.

To summarize, computing a network measure – degree, strength or directionality – on a given network yields a vector, e.g. $\vec{k} = (k_1, \ldots, k_N)$, holding the network measure score for each of the nodes.[4] In our spatial networks, these vectors can be interpreted as *spatial fields*, assigning a value to each point in the domain. We proceed to compare two climate networks by comparing these vectors using κ-statistics. Tab. 3.1 shows the different combinations of climate variable, network type, and network measure used in this study.

Table 3.1.: Overview of all combinations of climate variable, network type, and network measures appearing in this study.

Network	Climate variables	Time series	Similarity measure	Network measures
TCN	T_{mean}, T_{min}, T_{max}	Anomaly	Pearson's correlation r_{ij}	k_i, s_i
ESN	P	Binary event	Event synchronization e_{ij}	k_i, s_i, (ρ_i, φ_i)

3.3. κ-statistics for spatial agreement

We investigate the relationships between the N nodes in our simulated data and the basis data using network measures as proxies. Our approach is to focus on *spatial structures* found in the network measure scores.

As mentioned above, the spatial variability of STARS-generated data is restricted through the resampling process, and we expect the spatial distribution of network measure scores to reflect this property. Following the work of Kücken et al. we use κ statistics to quantify the agreement of spatial structures [35].

3.3.1. Categorization of network measures

Since the following steps rely on categorical data, we first convert the raw network measures $\vec{k}$ to categorical values $\vec{k}'$. The categorization is designed to be very simple and predictable: we are interested in those information contained in the networks and do not want to add additional layers of complexity at this stage.

[4] Some sample distributions of these vectors can be found in Fig. C.13.

We assign a node i to the category l if its network measure score k_i lies within the P_lth percentile of all scores in $\vec{k}$ using

$$P = \{1, 10, 25, 75, 90, 99, 100\}. \tag{3.20}$$

In other words, if $k'_i = l$ then $k_i \geq k_j$ holds true for P_l percent of all other nodes $i \neq j$. An example of some network measure data categorized in this way is given in Fig. B.2, and more details on the process can be found in App. B.2.

3.3.2. Cohen's original κ-statistic

Let ${}^a\vec{k}' = ({}^ak'_1, ..., {}^ak'_N)$ and ${}^b\vec{k}' = ({}^bk'_1, ..., {}^bk'_N)$ denote the network measures created from the datasets a and b, categorized into c categories.

The basic concept of the κ-statistic is to take the proportion of nodes that were placed into the same category in ${}^a\vec{k}'$ and ${}^b\vec{k}'$, called the proportion of agreement observed p_{O}, and compare it to the proportion of agreement expected by chance p_{C}, given the distribution of categories in both vectors.

The basic entity in the calculations is the probability p_{mn} of finding a node i in category m in a, and in category n in b, i.e. ${}^ak'_i = m$ and ${}^bk'_i = n$. For N nodes and c categories, this is given by

$$p_{mn} = \frac{1}{N} \sum_{i=1}^{N} \delta({}^ak'_i, m)\ \delta({}^bk'_i, n), \tag{3.21}$$

with

$$\delta(m, n) = \begin{cases} 1, & \text{if} \quad m = n \\ 0, & \text{else.} \end{cases} \tag{3.22}$$

The values of p_O and p_C can be obtained very efficiently with the help of a contingency table, or confusion matrix (see Tab. 3.2), as follows:

$$p_O = \sum_{m=1}^{c} p_{mm} \qquad \text{and} \qquad p_C = \sum_{m=1}^{c} p_{m\bullet} \cdot p_{\bullet m}, \tag{3.23}$$

where $p_{m\bullet} = \sum_n p_{mn}$ and $p_{\bullet n} = \sum_m p_{mn}$ are the so-called marginal distributions.

Cohen asserts that p_O will exceed p_C to the extend that 'nonchance factors are operating in the direction of agreement'. He defines

$$\kappa = \frac{p_{\mathrm{O}} - p_{\mathrm{C}}}{1 - p_{\mathrm{C}}}, \tag{3.24}$$

so that κ is equal to the 'proportion of agreement *after* chance agreement is removed from consideration' [36].

Table 3.2.: Generic contingency table of two vectors containing categorical values. There are c categories, so the table is a $c{\times}c$-matrix. The values in the 'total' row and column are also known as the table's 'marginal distributions' and are used to compute κ values.

	${}^{a}\vec{k}'$				
${}^{b}\vec{k}'$	1	2	$\cdots$	c	Total
1	p_{11}	p_{12}	$\cdots$	p_{1c}	$p_{1\bullet} = \sum_n p_{1n}$
2	p_{21}	p_{22}	$\cdots$	p_{2c}	$p_{2\bullet} = \sum_n p_{2n}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
c	p_{c1}	p_{c2}	$\cdots$	p_{cc}	$p_{c\bullet} = \sum_n p_{cn}$
Total	$p_{\bullet 1} = \sum_m p_{m1}$	$p_{\bullet 2} = \sum_m p_{m2}$	$\cdots$	$p_{\bullet c} = \sum_m p_{mc}$	1

The value range of Cohen's κ is $-\varepsilon \leq \kappa \leq 1$, where the lower bound $\varepsilon \geq 0$ depends on the marginal distributions involved. Importantly, κ is equal to one for complete agreement, and equal to zero when the observed agreement can be accounted for by chance. Negative values indicate worse than chance agreements.

It is possible to introduce a weighted version of κ (see App. B.3). If the categories follow some kind of order, the weighted κ_w can punish a disagreement involving two 'distant' categories more severely. The weighted version of κ has been implemented for this study (Lis. B.1), but was found not to influence the outcomes in any significant way. As a result, we proceeded to use the unweighted version for simplicity.

3.3.3. Pontius' $\kappa_{\mathbf{loc}}$ for categorical maps

We will apply κ to compare network measures computed from climate networks, i.e. scalar fields represented by vectors $\vec{k}$ of length N. We can plot such a field on a geographical map, and color the domain covered by each meteorological station (node) according to the value of the measure. We call data presented in this way a *cellular map*, since the coloring is done on irregular-shaped cells, corresponding roughly to each station's Voronoi polygon. If the raw network measure values $\vec{k}$ are categorized, this type of plot becomes a so-called *categorical map*, as seen in Fig. 3.3.

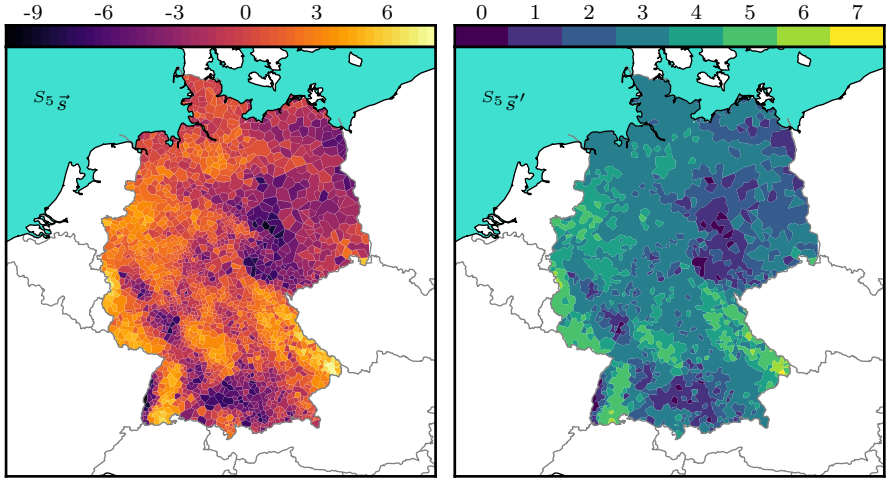


Figure 3.3.: Node strength on an event synchronization network calculated from STARS-generated winter precipitation data between 1700 and 1750. To the left the original data, to the right the categorized data.

Pontius pointed out two problems when applying Cohen's κ to categorical maps [37]:

First, the underlying chance-model of κ is only applicable if the marginal distributions on both maps are fixed. Should, however, the scientist have control over the marginal distributions, Cohen's κ might lead to wrong conclusions. In this study the marginal distributions are set by us, since the categorization of the raw values is based on percentiles.

Second, κ confounds errors in quantification with errors in location. When comparing two maps, both might be largely equal when it comes to the spatial distribution of categories (small location error), while diverging significantly on the overall quantity of cells belonging to each category (large quantification error). In this case, Cohen's κ might still be close to one, while giving no indication to the size of either error component.

Pontius proposed to replace Cohen's κ with κ_{loc} and κ_{quan}, designed to judge the agreement of two categorical maps by their agreement with regards to the locations of categories, and the sizes of categories, respectively.

We will focus solely on κ_{loc} since the sizes of our categories remain constant. In all ${}^{a}\vec{k}'$, category l includes the top P_l percent of N nodes, and consequently κ_{quan} is always close to one. The spatial distribution of these categories is, however, determined by the data. κ_{loc} is defined as

$$\kappa_{\text{loc}} = \frac{p_O - p_C}{p_M - p_C}, \tag{3.25}$$

with

$$p_M = \sum_{m=1}^{c} \min(p_{m\cdot}, p_{\cdot m}) \tag{3.26}$$

denoting the maximum possible proportion of agreement given the total count of cells in each category. It measures how well two maps agree with regard to the

spatial distribution of categories. An illustration of different κ_{loc}-values is given in Fig. 4.1. More about the decoupling of location and quantification errors in App. B.3.

Since we only use κ_{loc} and neither the original κ nor κ_{quan}, we will omit the subscript in the following sections. Instead, we use the subscript to denote the two data sources of the networks that are compared:

$$\kappa_{l,m} = \kappa(C_l, C_m), \tag{3.27}$$

with $l, m \in \{1, \ldots, 11\}$. For convenience, we also introduce as a short notation for the κ-score between a STARS chunk S_l and B

$$\kappa_l \equiv \kappa_{BS_l}. \tag{3.28}$$

3.4. Significance estimation

The statistical significance of κ scores is not always apparent. While in some contexts, a score of 0.8 might be a very significant result, in other cases the system under scrutiny has a very high base κ due to the structure or statistical properties of its data.

We will approach this problem from two angles. First, we will investigate whether our system has a substantial baseline level of κ with the help of reshuffled time series. Second, we will analyze how sensitive our methodological tool chain reacts to random noise added to the original time series.

3.4.1. Surrogate time series

Our study works with datasets that have a particular structure, consisting of N time series $x_i(t)$ of length n, each covering a fixed area v_i around the location $(\varphi, \lambda)_i$. Each time series contains observations of a climate variable x, which implies certain boundaries for the value range, e.g. $x \geq 0$ for precipitation, its statistical characteristics such as $\langle x \rangle$ and σ_x, and its frequency distribution $p(x)$. This is true both for B and S_l, since STARS is designed to ensure these physical characteristics in its simulated time series (see App. A.1).

These features might result in a baseline κ inherent to the system that is different from zero. For example, networks computed from precipitation data could always resemble each other in some way simply because the raw data has a certain characteristic distribution. Another possible source of bias is due to the STARS mechanism: as explained in Sec. 2.1.2, STARS never mixes the contents of two time series. It generates the simulated time series at node i using only data from that location. Moreover, the resampling is done synchronously at all locations.

In order to test for these possible biases, we generate shuffled time series designed to still retain the main statistical characteristics of our original time series. The networks calculated from these time series, and the network measures derived from them, are used to investigate the system's baseline κ. To generate the surrogate time series C_l^s for a STARS chunk S_l, the data in S_l were shuffled for each node

individually, respecting the appropriate month of the year. Let $x_i(t)$ denote the time series at node i. We collect all observations $x_i(t_m)$, $t_m \in \{t_1, \ldots, t_n\}$, that belong to the same month M into a common pool:

$$\text{pool}_M = \{x_i(t_m) : \text{month}(x_i(t_m)) = M\}, \quad M = \text{Jan}, \ldots, \text{Dec}. \tag{3.29}$$

A pool has thus between 50×28 and 50×31 elements. Each pool is shuffled at the level of individual days, and a new time series is assembled from the twelve shuffled pools (see Fig. B.7 and Lis. B.2). Importantly, we shuffle at each nodes individually, instead of all nodes synchronously as does STARS.

The key features of the surrogate time series are as follows:

1. At each node, they conserve $p(x_i)$, some statistical properties such as $\langle x_i(t) \rangle$, some aspects of the annual cycle, and the approximate value range of $x_i(t)$.

2. They also conserve the same characteristics for the overall field of all N nodes.

3. They do not conserve temporal correlations between nodes beyond those rooted in the annual cycle.

For each of the N nodes, and each of S_l and B, we generated an ensemble of 1000 surrogate time series S_l^s and B^s. From these surrogate time series we constructed the networks in the same way as the original networks were generated from the original STARS and basis time series. We expect any κ calculated on those surrogates to reflect spatial patterns in $\vec{k}$ that arise from the first two features while being unaffected by the temporal correlations between time series.

Figs. B.5 and B.6 are representative for B^s and S_l^s, showing that the surrogate κ-values have a normal distribution with a relatively sharp peak. The distribution means $\langle \kappa(C_l^s, C_m) \rangle$ for the different combinations of variable, network, density and measure lie between 0.05 and 0.20. On average, they are close to 0.10 with a standard deviation of around 0.01.

These numbers imply that the data type and structure does indeed introduce a bias to κ, although the effect is rather small at around 0.10.

Since the surrogate κ-values are roughly normally distributed, we can also calculate the z-score indicating how far above or below the ensemble mean a particular observation lies, in units of one standard deviation:

$$z(C_l, C_m) = \frac{\kappa(C_l, C_m) - \langle \kappa(C_l^s, C_m) \rangle}{\sigma_{\kappa(C_l^s, C_m)}}, \tag{3.30}$$

where $l, m \in \{1, \ldots, 11\}$, $\kappa(C_l^s, C_m)$ denoting the 1000 κ-values calculated between the surrogates of C_l and C_m, and $\langle \cdot \rangle$ and $\sigma_{\cdot}$ the mean and standard deviation of these values. Given that all the surrogate distributions are very similar, we save computational resources by using B^s and calculating

$$z(C_l, C_m)^* = \frac{\kappa(C_l, C_m) - \langle \kappa(B^s, C_l) \rangle}{\sigma_{\kappa(B^s, C_l)}}, \tag{3.31}$$

for all $\kappa(C_l, C_m)$ in this study to test for significance. Using $z(C_l, C_m)^*$, we find our κ-values to be significant at the 10-σ-level and far above, with the mean z-scores being around 65 for temperature, and 40 for precipitation data (Figs. B.4 and B.3). The STARS-generated data are thus in significantly higher spatial agreement with the basis data then the more random, less physical surrogates.[5]

3.4.2. Sensitivity to random noise

Since our methodology is complex and non-linear, we conducted additional tests on the susceptibility of our κ-scores to random noise added to the time series. It is not immediately clear how such changes to the raw data propagate through our methodological tool chain.

We chose a small subset of $N' = 441$ nodes and time series of length $n' = 1000$, corresponding to the summer data between 1 June 1650 and 19 August 1660, from the STARS dataset for a preliminary test. We calculated networks from the original values, and 100 times from copies with added uniformly distributed noise. For the observation $x_i(t_m)$ we obtained the noisy observation $x_i^n(t_m)$ as follows:

$$x_i^n(t_m) = x_i(t_m) + \delta. \tag{3.32}$$

The noise δ is generated using the probability density function of the uniform distribution,

$$p(\delta) = \frac{m}{2\,|x_i(t)|}, \tag{3.33}$$

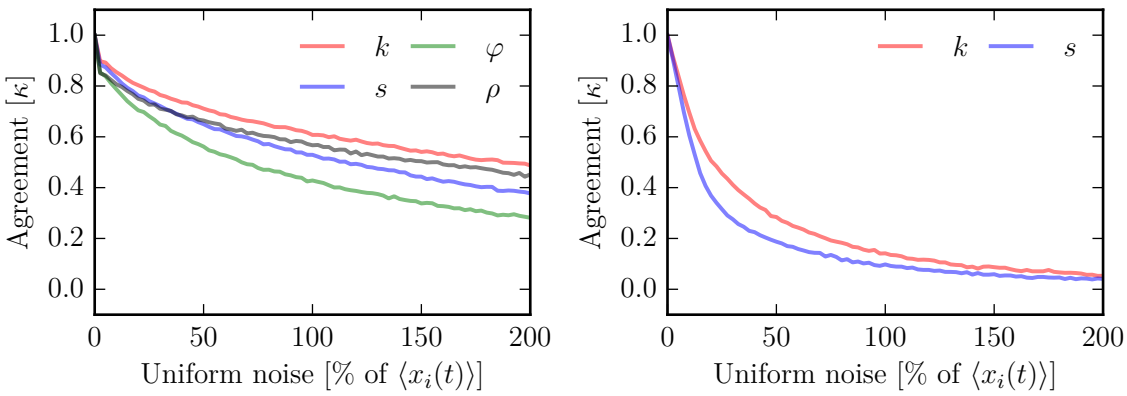
where $m \in [0, 2]$ is a parameter determining the magnitude of noise.

On each network, we calculated the degree k_i of all nodes, and compared the categorized $\vec{k}'$ between the original and the noisy data with κ. The results, averaged over the 100 realizations of the noisy time series, are shown in Fig. 3.4.

In the small test networks, κ responds immediately to added noise: if the noise is in the order of five percent of the time series mean, κ drops to around 0.9, and an additional five percent drop it to 0.8. We find that TCNs are more sensitive to noise, and that degree generally is more robust than the other measures.

We extended this test to the full dataset by adding uniformly distributed noise according to Eq. (3.32) to the observations in S_l and B. In Fig. 3.5, we see κ_l for all chunks, comparing the STARS data, STARS data with different levels of noise (1 realization), and the surrogates as described in Sec. 3.4.1.

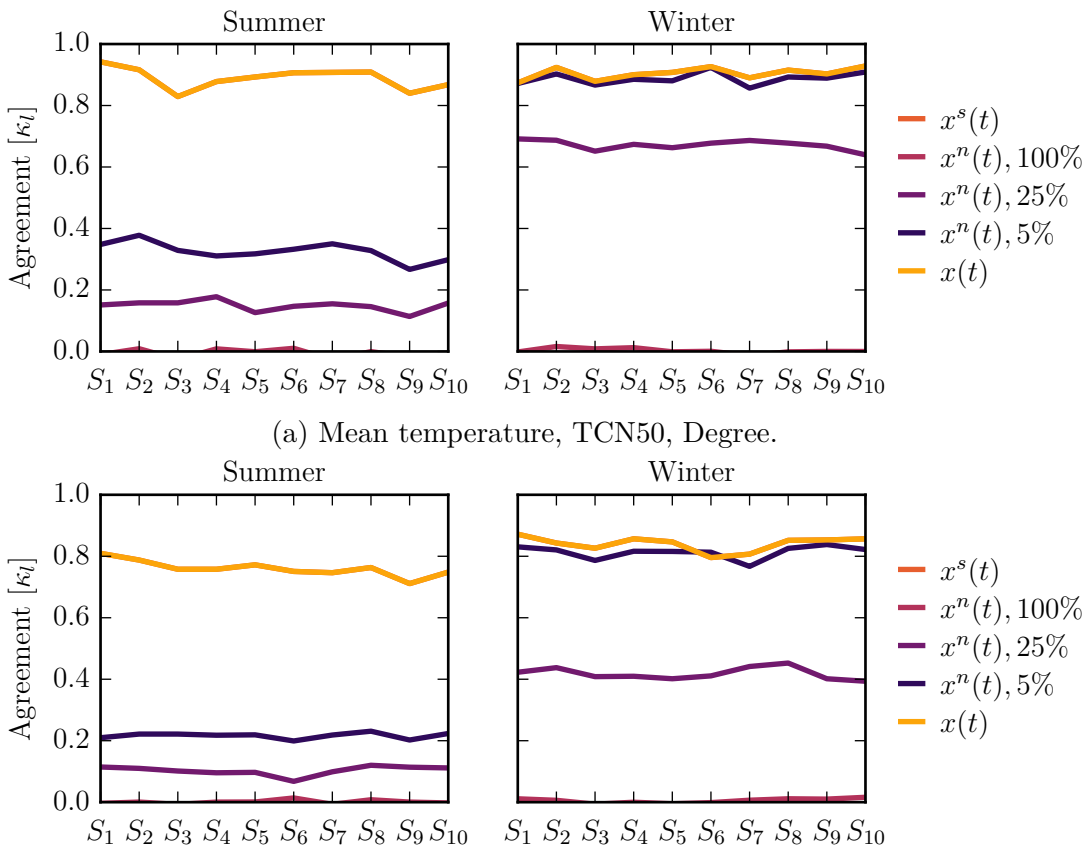
[5]Some studies such as [31] completely forgo the reporting of the raw κ-values in favor of z-scores, which have the advantage of giving a direct indication of significance. Our surrogates, however, are unsuitable for this approach, since they contain essentially random time series, albeit respecting some characteristics of climate data. Importantly, they result in very low $\sigma_{\kappa(C_l^s, C_m)}$, which leads to high variability of the z-scores. In other words, a z-score of 40 in our case is not necessarily indicative of a much higher significance than a score of 20, but instead mainly due to variations of the very small denominator in (3.30).



(a) Precipitation (b) Mean temperature

Figure 3.4.: Propagation of noise through the methodological tool chain to the κ-statistic. For degree, $\varrho = 50\%$ was used.

For strength on TCNs calculated from temperature data, STARS exhibits an average spatial agreement of around 0.8 to 0.9 in both seasons, the winter agreements being slightly higher and closer to 0.9 on average. This puts the STARS data in the same range as noisy data at the 5%-level. The same is true for degree at 50% link density, where the average κ-level is a little lower around 0.8. There are no major differences between the different temperature variables analyzed (Fig. 3.5).



(a) Mean temperature, TCN50, Degree.

(b) Mean temperature, TCN, Strength

Figure 3.5.: STARS performance compared to surrogates and noisy time series as measured with κ_l.

The sensitivity to noise is slightly different between seasons: for strength, the winter data in general show a slower drop-off in spatial agreement when noise is added, with the 25%-level still at just below 0.7 and the 100%-level between 0.2 and 0.4. For summer data, in comparison, the κ drops off quicker, with the 25%-level being at 0.3 or below. κ drops faster with higher noise levels for link density than strength.

Turning to precipitation, the overall agreement level is around 0.7 for degree at 50% link density both in summer and winter, between 0.3 and 0.6 for strength in the summer, and around 0.6 in the winter. For directionality ρ, we have a general level of 0.3 in the summer and 0.5 in the winter. For φ, the agreement is around 0.3 to 0.6 in the summer, with a large increase after S_3, and between 0.6 and 0.8 in the winter (Figs. B.8 and B.9).

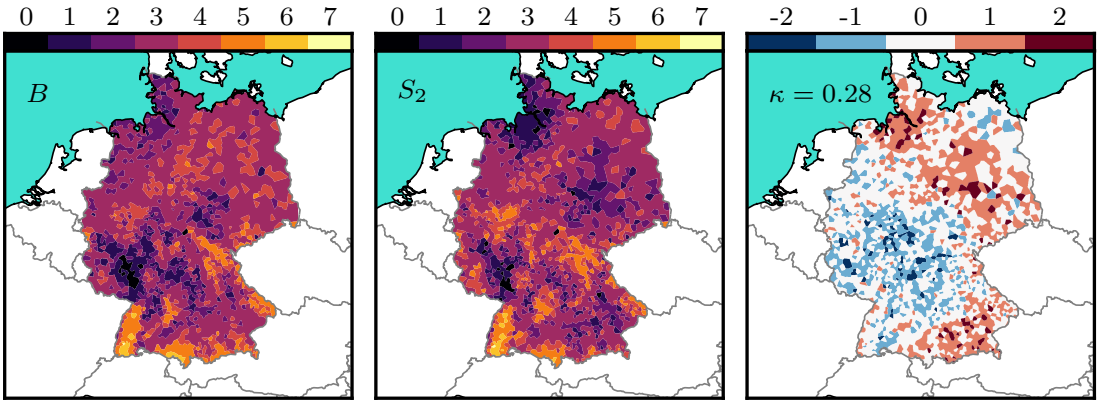
When working with ESNs, κ remains relatively unaffected by added noise. This comes as no surprise since the event synchronization score e only changes where noise pushes an observation above or below the 10mm static threshold. The lines for all levels of noise are thus very close to the STARS line, only at the 100%-level can we see an overall κ decrease of around 0.1. This is true for all measures, degree, strength and directionality.

To summarize, κ-scores computed between ESNs are quite resistant to random noise added to the raw data. This is mainly a result of the use of event series $\hat{x}(t)$ that effectively eliminates much of the impact that noise has on $x(t)$. Our TCNs are based on Pearson's correlation coefficient and are much more sensitive to noise. This implies that variations of κ-scores between chunks are likely more significant in the case of ESNs than TCNs, where they might occur more often due to random effects.

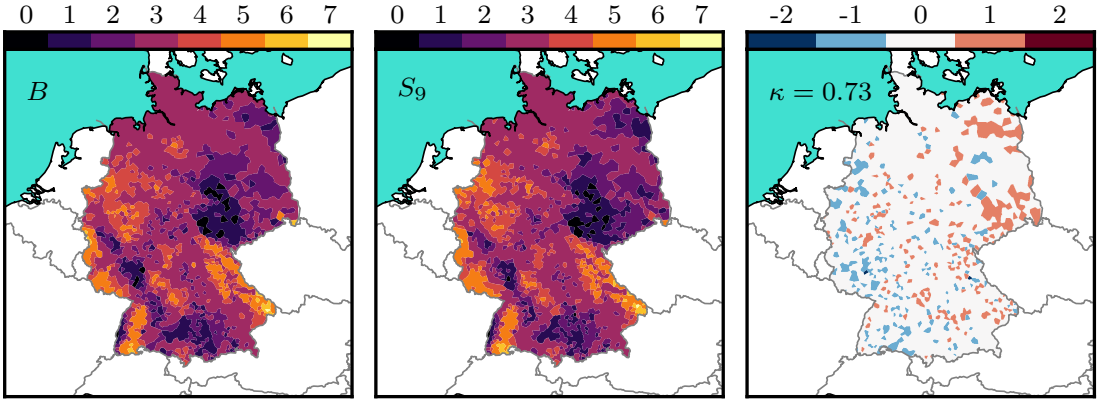
4. Analysis

Our κ-scores measure agreement between the spatial patterns in climate networks that were generated from a chunk C_l of our STARS or basis data. A perfect agreement, i.e. the spatial distribution of network measures is the same in two chunks, would result in $\kappa = 1$. Agreement that is a product of mere chance would result in $\kappa = 0$.

An intuition for how these values translate to categorical maps of our climate networks can be gained with the help of Fig. 4.1. In particular, even two maps with a relatively low κ-score of 0.28, as seen in Fig. 4.1a, seem to share some prominent features such as general locations of higher and lower values.



(a) P, ESN, Strength. Example of low spatial agreement ($\kappa = 0.28$) between two networks.



(b) P, ESN, Strength. Example of high spatial agreement ($\kappa = 0.73$) between two networks.

Figure 4.1.: Comparison of high and low spatial agreement between networks as measured by κ. The rightmost plot shows the difference in the category of a particular cell between both maps.

4.1. Comparison of STARS and basis data

The first step in our analysis is to examine the agreement between the STARS chunks S_l and the basis chunk B, i.e. the agreement of the simulation data with the observation data κ_l. The average κ_l in our study is 0.73 ± 0.17 (see Tab. 4.2).

4.1.1. Analysis by chunk time period

In the following, we will analyze the development of κ_l over time, in order to investigate whether STARS data grows increasingly similar – or dissimilar – to the input data the closer both are in time. Figs. 4.2 and 4.3 show the results for node degree at 50% link density and node strength, respectively.

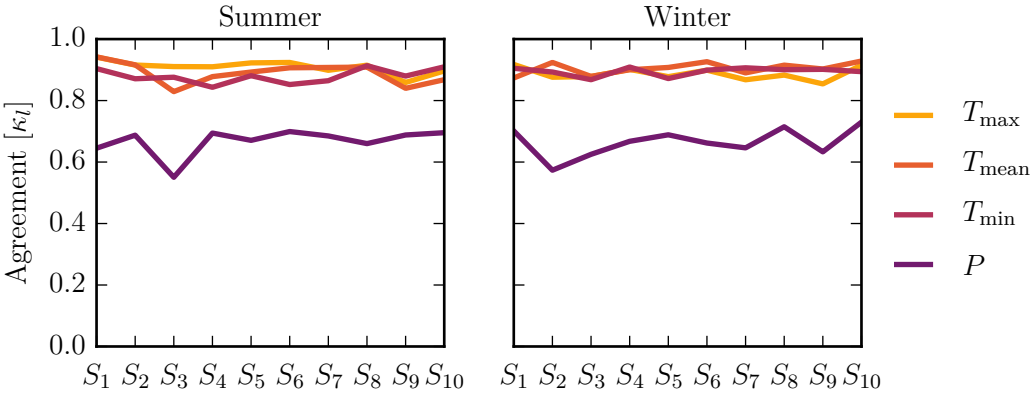


Figure 4.2.: TCN50 and ESN50, Degree. Development of κ_l over time.

For the three temperature variables, no trend is visible. Instead, the spatial agreement with B, as measured by κ_l, is relatively stable at around 0.9 ± 0.02 for degree, and 0.8 ± 0.03 for strength (see Tab. 4.1). There are some rather small fluctuations for strength, for example a small decrease in the agreement for strength for the chunks S_5 and S_6 in the winter, an an overall slight increase in agreement for the last chunk S_{10} that covers basically the same time period as B.

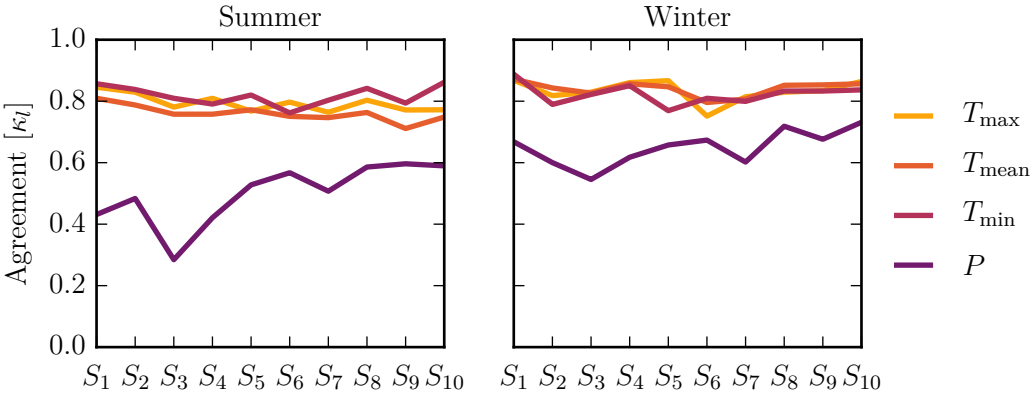


Figure 4.3.: TCN and ESN, Strength. Development of κ_l over time.

Looking at the precipitation data, the variability of agreement is higher, especially for node strength, while the average agreement is lower. In Tab. 4.1 we see that σ_{κ_l} is around 0.03 for the temperature variables, and between 0.04 and 0.09 for

precipitation. We will discuss the overall lower level of agreement later. Visually, we find that there is no trend for node degree, but one might argue the presence of an upwards trend when looking at the strength curves in 4.3. There is a noticeably lower agreement for S_3 in all but the strength winter plot.

Generally speaking, we find that the peaks and valleys of the curves are not in sync across climate variables. The dip for S_3, for example, is only visible for precipitation, and to some degree in the summer degree curve for mean temperature, but not in any of the other curves in those figures. This, together with the small magnitude of fluctuation that we see, suggests that the variations are more likely to be random than due to a systematic effect within the data. An exception is the slightly higher agreement of the last chunk S_{10} that is shared across all climate variables.

The directionality measure was only computed for precipitation data, the results can be seen in Fig. 4.4. For φ, the average κ_l is 0.49 ± 0.11 and 0.72 ± 0.05 in summer and winter, respectively. The directionality strength has averages of 0.34 ± 0.05 and 0.56 ± 0.06 for these seasons, and thus the lowest agreement of any network measure. We again notice a slight upward trend, and both curves seem to match their general development. For example, there is a common increase in agreement after S_3 in the summer, and a dip in agreement for S_8 in the winter.

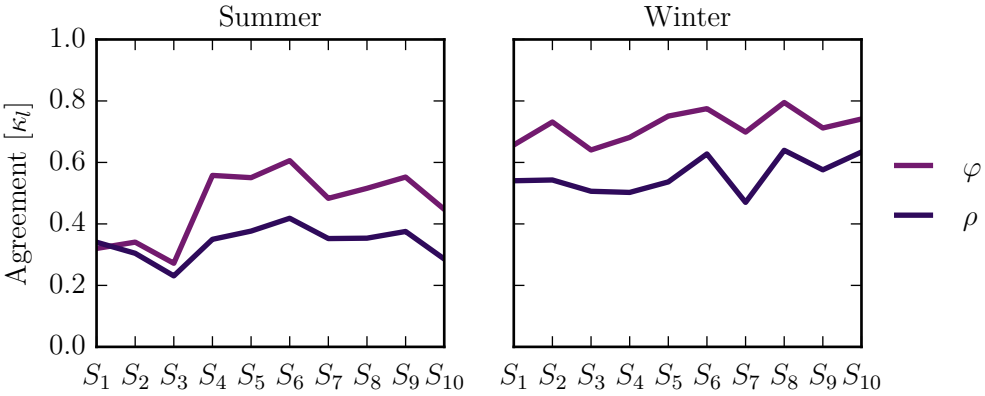


Figure 4.4.: ESN, Directionality. Development of κ_l over time.

Tab. 4.1 summarizes the results in terms of mean κ_l scores and their standard deviation. In addition, we carried out a linear regression for each combination of climate variable, network measure, and so on. In other words, we fitted

$$\kappa_l = a \cdot l + b \tag{4.1}$$

with $l = 0, \ldots, 10$. Note that in the case of node degree k, the table only gives the results for a link density of 50%, the remaining results can be found in Tab. C.3. We find that the slope a is consistently close to zero, even for the precipitation data. At the same time, the coefficients of determination r^2 for these regressions are below 0.6 for all cases, and for most even below 0.2. Concerning the slight upwards trend that we detected in the precipitation data, especially considering node strength (Fig. 4.3) and directionality (Fig. 4.4), the results from these regressions seem to indicate that the effect is small at best.

Table 4.1.: Mean and standard deviation of κ_l per network measure. The last columns give the coefficients of a linear regression of $\kappa_l = a \cdot l + b$ and the respective coefficient of determination r^2.

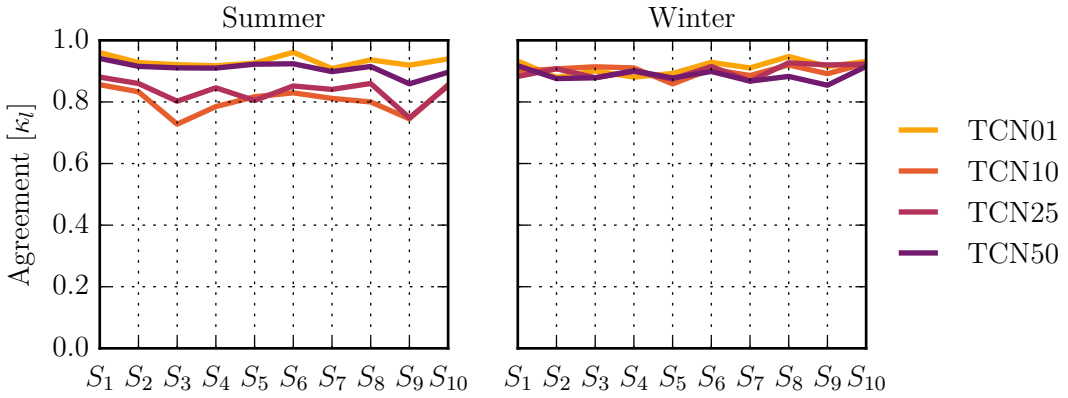
Measure	Season	Variable	Link density	$\langle\kappa_l\rangle$	σ_{κ_l}	a	b	r^2
k	Summer	T_{max}	50	0.91	0.02	-0.005	0.931	0.46
		T_{mean}		0.89	0.03	-0.004	0.908	0.13
		T_{min}		0.88	0.02	0.002	0.870	0.07
		P		0.67	0.04	0.006	0.640	0.17
	Winter	T_{max}		0.89	0.02	-0.001	0.893	0.04
		T_{mean}		0.90	0.02	0.003	0.891	0.24
		T_{min}		0.89	0.01	0.001	0.891	0.04
		P		0.66	0.04	0.006	0.636	0.16
s	Summer	T_{max}		0.79	0.03	-0.006	0.823	0.49
		T_{mean}		0.76	0.02	-0.007	0.791	0.61
		T_{min}		0.82	0.03	-0.001	0.821	0.01
		P		0.50	0.09	0.024	0.390	0.56
	Winter	T_{max}		0.83	0.03	-0.001	0.838	0.01
		T_{mean}		0.84	0.02	-0.001	0.845	0.01
		T_{min}		0.82	0.03	-0.001	0.829	0.01
		P		0.65	0.05	0.012	0.596	0.38
φ	Summer	P		0.46	0.11	0.022	0.364	0.34
	Winter	P		0.72	0.05	0.009	0.678	0.29
ρ	Summer	P		0.34	0.05	0.004	0.321	0.05
	Winter	P		0.56	0.06	0.010	0.510	0.28

To summarize, the κ_l in the precipitation data is generally lower, with a somewhat higher variability, than in the temperature data. The three temperature curves stay relatively close to a value of $\kappa_l \approx 0.85$, with little variation. Furthermore, we visually detected a slight upwards trend over time in the precipitation data, but this trend could not be backed by the results of our linear regressions, where we only find $0.2 \leq r^2 \leq 0.4$ for the data in question. We find that curve features are generally not shared across climate variables, so a slightly higher κ_l for one variable does not imply the same for another variable.

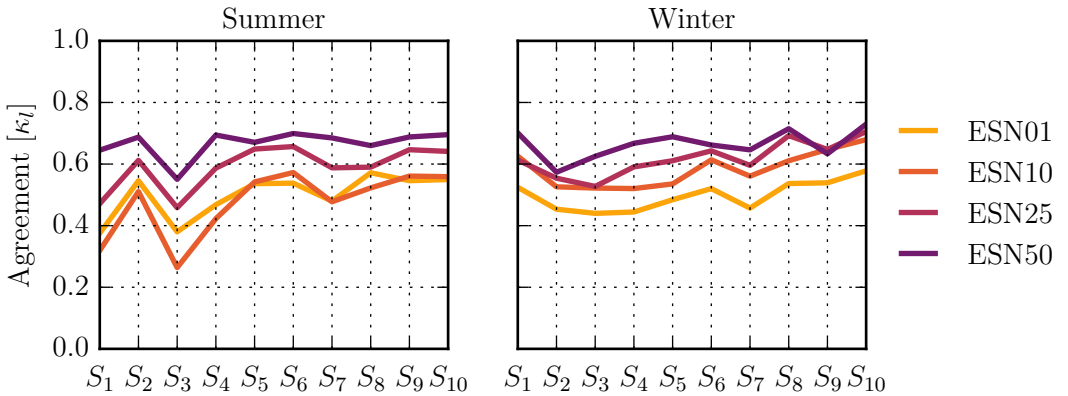
4.1.2. The effect of link density

For the calculation of node degree, we convert link-weighted networks to unweighted networks via thresholding. We chose it so that the resulting $\mathcal{A}$ exhibits a prescribed link density ϱ, as described in Sec. 3.1.2. Since ϱ and network measure scores are connected, we can establish some level of comparability by using similar link densities, even between different network types. Typical thresholds that lead to densities of 1% and 50% are around 0.99 and 0.85 for TCNs and 40 and 15 for ESNs, respectively.

From Eq. (3.10) it is obvious that ϱ affects the absolute values of node degree: more links mean more elements in the neighborhood, leading to higher degree values at each node. For example, the mean degree is around 5 in our lowest-density networks, and above 20 for the highest-density ones. In our methodology, a low link density is equivalent to a high threshold of the similarity measure, in other words only the most strongly correlated node pairs are taken into account in low-density networks.



(a) $T_{\max}$, TCN, Degree.



(b) P, ESN, Degree.

Figure 4.5.: Influence of link densities in unweighted networks on κ_l. The effect is stronger for precipitation data, with the highest link densities generally having the highest agreements. $T_{\max}$ is representative for other temperature variables.

We find that for precipitation, lower densities tend to lead to lower κ-scores, an effect that is on the order of 0.1 to 0.2 of the absolute value (Fig. 4.5). In other words, if our networks consider only the most important relationships, then the relative importance of the nodes in the network varies more strongly between STARS and basis data. In this sense, STARS might be said to conserve the overall spatial pattern of 'baseline correlation' between nodes, while changing the locations of the 'top correlations'. For temperature data, the effect is not clearly visible.

These findings might, however, be primarily of statistical origin. Comparing S_l with B, we find that STARS approximately conserves the frequency distribution of correlations of its input (Fig. 4.6). While these are not normal distributions, they have a pronounced peak. There are thus many more relationships that fall in some medium range than those close to the extreme values, i.e. there are vastly more node pairs 'somewhat correlated' than there are pairs 'not related at all' or 'very closely related'. If we were to conserve this frequency distribution, but randomize it spatially, we would statistically often replace a mid-range value with another mid-range value, but only very rarely replace a high value with another high value. In other words, it is less likely to reproduce the spatial distribution of the extremes by chance. Since the distribution of e has a sharper peak than that of r, we expect the effect to be stronger for precipitation data.

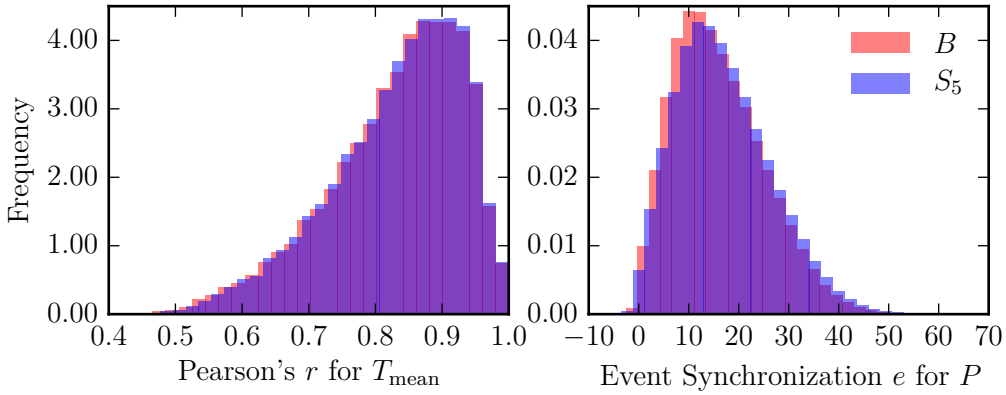


Figure 4.6.: Frequency distributions for similarity measures used as a basis for network construction.

In Fig. 4.5, we can see that generally, the curves for a particular climate variable share common features for different link densities, i.e. there are peaks and valleys at the same locations. There are some exceptions, for example a small dip in the winter precipitation curves at S_9 that is shared between ESN25 and ESN50, but not ESN01 and ESN10. We thus conclude that one link density is somewhat representative for the other densities, although a more thorough exploration of the connection between link density and frequency distribution, and the effects on κ, might be interesting for the future.

4.1.3. Analysis by chunk mean temperature

Having found no indications of that would indicate that there are indeed long-term trends visible so far, we try to look at the data using a different perspective. The overall mean temperature of our climate data is of special significance in our study. As described in Sec. 2 and App. A.1, temperature trend lines were the only forcing parameters in the STARS resampling scheme. Consequently, temperature is the only independent variable in this system. It is thus interesting to look at the data ordered by chunk seasonal mean temperature $\langle T_l(\mathcal{S})\rangle$.

Let $x_i(t)$ denote the time series of T_{mean} at node i, then we obtain the mean temperature of chunk C_l for season $\mathcal{S}$ by averaging over all nodes and all relevant time steps:

$$\langle T_l(\mathcal{S})\rangle = \frac{1}{N}\sum_{i\in\mathcal{N}}\frac{1}{n}\sum_{t\in\mathcal{T}} x_i(t) \tag{4.2}$$

with $\mathcal{N}$ the set of N nodes and $\mathcal{T} = \{t_m : t_m \in C_l \ ; \ t_m \in \mathcal{S}\}$ the set of n time steps that fall within C_l, $l \in \{1, \ldots, 11\}$, and season $\mathcal{S}$. This, and the resulting order of chunks, is summarized in Tab. 2.2.

Figs. 4.7 and 4.8 show degree and strength for mean temperature and precipitation ordered in this way, the corresponding figure for directionality (Fig. C.1) can be found in the appendix. Note that S_{10} covers roughly the same time period as B and thus $\langle T_{10}\rangle \approx \langle T_{11}\rangle$.

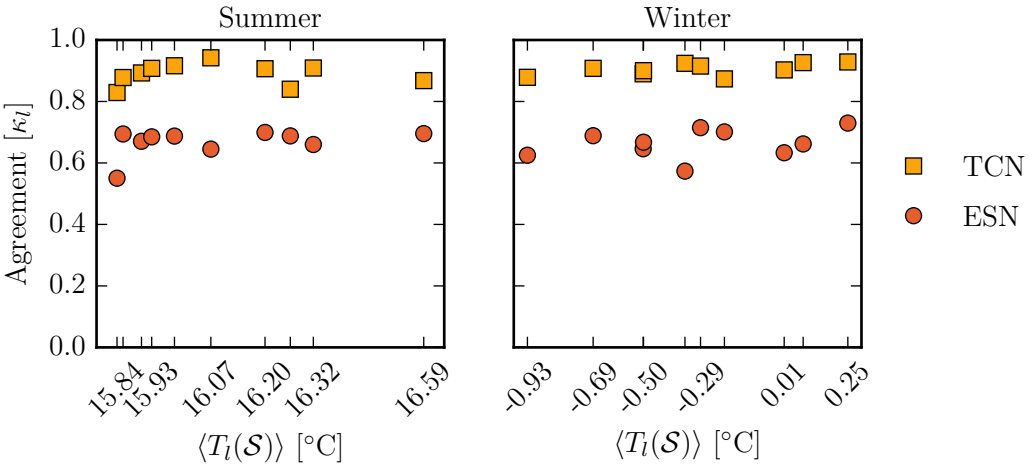


Figure 4.7.: T_{mean} and P, TCN50 and ESN50, Degree. Development of κ_l as a function of chunk mean temperature.

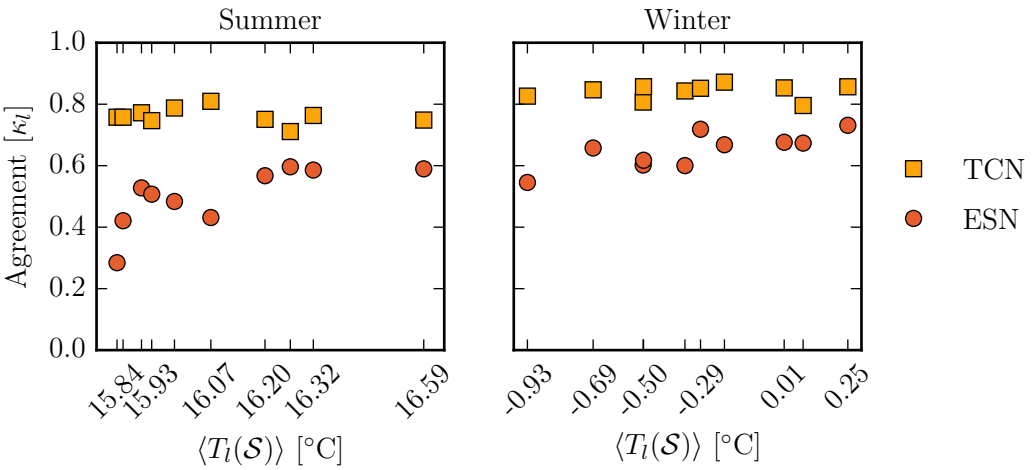


Figure 4.8.: T_{mean} and P, TCN and ESN, Strength. Development of κ_l as a function of chunk mean temperature.

For mean temperature, neither degree nor strength show a tendency. For precipitation data, only a very slight upward trend can be seen for strength, and none for degree. The directionality measures ρ and φ are especially volatile in the summer, with no clear trend in either the summer or winter plots (see Fig. C.1). We again performed linear regressions on these curves, i.e.

$$\kappa_l = a \cdot \langle T_l(\mathcal{S})\rangle + b, \tag{4.3}$$

the results are summarized in Tabs. C.1 and C.2. We again find r^2-values to be generally below 0.4 and very small slopes a. The notable exception is node strength for our precipitation data (Fig. 4.8), where the slopes are $a = 0.29$ and 0.12 with $r^2 = 0.60$ and 0.53 for summer and winter, respectively. While these could be seen as significant, a visual inspection of the data shows that they are most likely to be a result of the exceptionally low agreement for the two coldest chunks when compared to B.

To summarize, even though temperature is the only independent variable of the STARS process, we find very little indications that the model produces spatially

more similar data when temperatures are closer to the input data. More precisely, κ_l does not seem to be systematically influenced by chunk mean temperature, although an argument can be made that for the combination node strength-precipitation data, such a tendency is visible.

4.2. Intercomparison of chunks

So far, we only analyzed the spatial agreement between S_l and B, and found no evidence of any trend in the development of κ_l over time or with chunk seasonal mean temperature. A possibility remains, however, that there are such trends inside the STARS dataset. For example, if the Little Ice Age is reflected in the data, and the peak of its effect is around 1600, we would expect a higher spatial agreement between the chunks around that time. Such a convergence *inside* the STARS dataset would not necessarily be visible in the κ_l-plots that we analyzed above.

To examine this hypothesis, we calculated the full set of $\kappa_{l,m}$ for $l, m \in \{1, \ldots, 11\}$. The results are 11×11-matrices that show the spatial agreement between all combinations of chunks and basis data for a given combination of climate variable, network and network measure. They are expected to reveal trends inside the STARS dataset when it comes to spatial patterns, for example convergence towards a particular pattern that was forced by a long-term trend. The average $\kappa_{l,m}$ in our study, with $l, m \in \{1, \ldots, 10\}$, is 0.69 ± 0.19 (see Tab. 4.2).

4.2.1. Analysis by chunk time period

The first analysis orders the chunks according to time, so that moving from left to right and from bottom to top in the matrix, time moves forward. If there are local developments inside the STARS dataset that reflect long-term trends, for example global warming associated with industrialization, we expect some visible features to emerge in the agreement matrices.

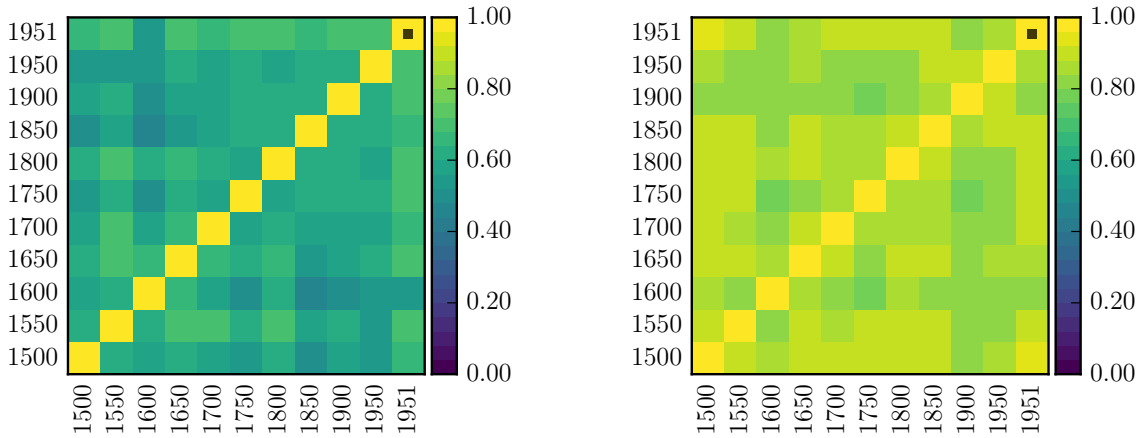
Such features could be visible as an overall higher level of agreement in a small cluster of cells inside the matrix, implying that a subset of the chunks resemble each other more than the average. For example, if the periods most affected by the Little Ice Age, C_2 and C_3, share common features, we would expect brighter spots for the respective matrix elements. Or, we could observe that agreement is generally growing or falling monotonically when moving from left to right, which would indicate that there is a spatial pattern towards which the simulation converges. Finally, if there were generally higher values close to the diagonal, we could conclude that chunks closer in time to each other tend to agree better.

Average agreement levels

We find that overall, agreement is significantly higher, 0.69 ± 0.19 on average, when using STARS data than when using the randomized data of our surrogates (also see Sec. 3.4.1). This confirms our expectation that STARS does generate output with somewhat consistent spatial patterns. Indeed, there seems to be a base spatio-temporal variability contained in the basis data that can be found in each STARS chunk, but that variability is not substantially increased or decreased during the

simulation period.

The matrices in Fig. 4.9 and 4.10 are representative examples for the other network measures, which can be found in the App. C.2.



(a) P, Summer, ESN50, Degree.

(b) T_{mean}, Summer, TCN50, Degree.

Figure 4.9.: Spatial agreement between all chunks, ordered by time. Each label is the starting year of each 50-year-chunk. The basis chunk B is marked by a black rectangle.

When comparing the temperature and precipitation, we find that precipitation data apparently exhibits more varied spatial patterns, as is evidenced by the overall lower agreement level in the matrices, $\kappa_{l,m} = 0.48 \pm 0.14$ instead of 0.82 ± 0.07. These differences are clearly visible in Fig. 4.9. Generally speaking, precipitation dynamics are extremely complex, and climate models routinely perform worse when handling precipitation data than other climate variables. For example, this was found to be true for STARS and a dynamical model in [8]. Using more complex difference metrics, Lange et al. later found that model performance is even worse over land than over sea [6], so these results are not surprising. Since STARS does not really model physical processes, we propose two possible explanations tied to the STARS mechanism.

First, the lower agreement for precipitation networks could indicate that the precipitation data generated by STARS indeed have a higher spatial variability than the temperature data. This could be a consequence of their non-Gaussian distribution that results in more drastic differences between chunks when resampled. If this were the case, STARS performance would actually depend on the climate variable, i.e. it might be better suited for the simulation of temperature data than precipitation, or vice versa. In turn, this calls into question the validity of some assumptions made in the STARS mechanism as described in App. A.1 and [1], notably the decision to resample only a single climate variable and co-develop all others without further consideration.

Second, the increased variability of precipitation networks could also follow from the differences between the similarity measures e and r which are at the basis of ESNs and TCNs, respectively. While Pearson's r exclusively measures linear correlations between nodes, the event synchronization score e captures non-linear relationships

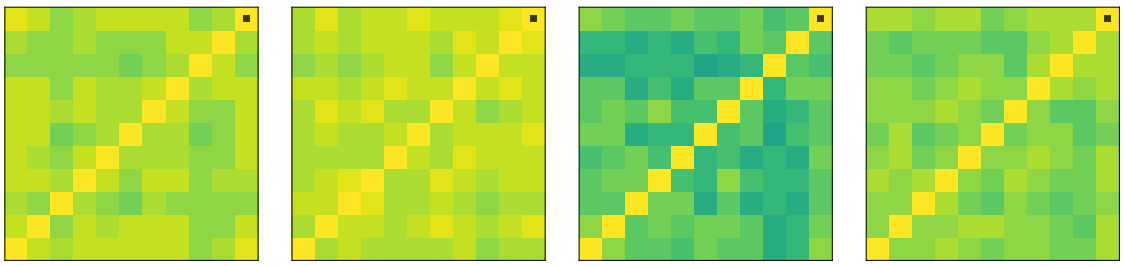
as well. In addition, e works on event time series $\hat{x}_i(t)$ that actually lose some of the information contained in $x_i(t)$, gaining a higher robustness to small disturbances in the process as found in Sec. 3.4.2.

Table 4.2.: Mean and standard deviation of κ_l and the STARS-only $\kappa_{l,m}$ with $l, m \in \{1, \ldots, 10\}$, grouped by various criteria. The filter column indicates the values included in the average, e.g. the s-rows include only κ-values for node strength, but not for other network measures.

Filter		$\langle\cdot\rangle$	$\sigma_\cdot$	Filter		$\langle\cdot\rangle$	$\sigma_\cdot$
-	κ_l	0.73	0.17	k	κ_l	0.80	0.15
	$\kappa_{l,m}$	0.69	0.19		$\kappa_{l,m}$	0.77	0.17
Summer	κ_l	0.68	0.20	s	κ_l	0.75	0.12
	$\kappa_{l,m}$	0.64	0.22		$\kappa_{l,m}$	0.71	0.14
Winter	κ_l	0.77	0.13	φ	κ_l	0.59	0.15
	$\kappa_{l,m}$	0.73	0.15		$\kappa_{l,m}$	0.54	0.15
TCN	κ_l	0.84	0.06	ρ	κ_l	0.45	0.12
	$\kappa_{l,m}$	0.82	0.07		$\kappa_{l,m}$	0.38	0.12
ESN	κ_l	0.55	0.13	$k,\ \varrho = 1\%$	κ_l	0.82	0.19
	$\kappa_{l,m}$	0.48	0.14		$\kappa_{l,m}$	0.80	0.21
T_{mean}	κ_l	0.84	0.06	$k,\ \varrho = 50\%$	κ_l	0.84	0.10
	$\kappa_{l,m}$	0.81	0.08		$\kappa_{l,m}$	0.81	0.12
T_{max}	κ_l	0.85	0.06	$k,\ \varrho = 25\%$	κ_l	0.80	0.12
	$\kappa_{l,m}$	0.82	0.08		$\kappa_{l,m}$	0.76	0.14
T_{min}	κ_l	0.85	0.05	$k,\ \varrho = 10\%$	κ_l	0.76	0.15
	$\kappa_{l,m}$	0.83	0.06		$\kappa_{l,m}$	0.73	0.17
P	κ_l	0.55	0.13				
	$\kappa_{l,m}$	0.48	0.14				

Simulation and observation data

The agreement of S_l with B, i.e. between STARS and the basis data, is generally similar to that of S_l with other S_m, i.e. within the STARS dataset. Although the matrices in Fig. 4.9 show slightly higher average agreement for the top row and last column, the absolute values seen there are not higher than those encountered in other parts of the matrices. Looking at Tab. 4.2, we see that $\langle\kappa_l\rangle = 0.73 \pm 0.17$ and $\langle\kappa_{l,m}\rangle = 0.69 \pm 0.19$, where $l, m \in \{1, \ldots, 10\}$, i.e. only include the STARS data. The effect is barely visible in matrices such as Figs. 4.10 and C.2.



(a) T_{mean}, TCN50, Degree. (b) T_{mean}, TCN, Strength.

Figure 4.10.: Spatial agreement between all chunks, ordered by time. For each subplot the left plot shows the summer and the right plot the winter data.

In other words, two STARS-generated datasets are likely to resemble each other to approximately the same extent as they resemble the data from which they were generated, provided that both datasets were sampled from the same input data. This strengthens the case for our methodological choice to split the original STARS dataset into chunks in order to perform our analyses by time, thus treating every chunk as a separate dataset.

Visible trends

Again referring to Figs. 4.10 and C.2, the matrices do not expose any clear trends of increasing or decreasing κ-scores over time. We can neither see rows of monotonically growing or falling values, nor clusters of substantially higher or lower agreement within the matrices, nor a generally different level of agreement close to the diagonal.

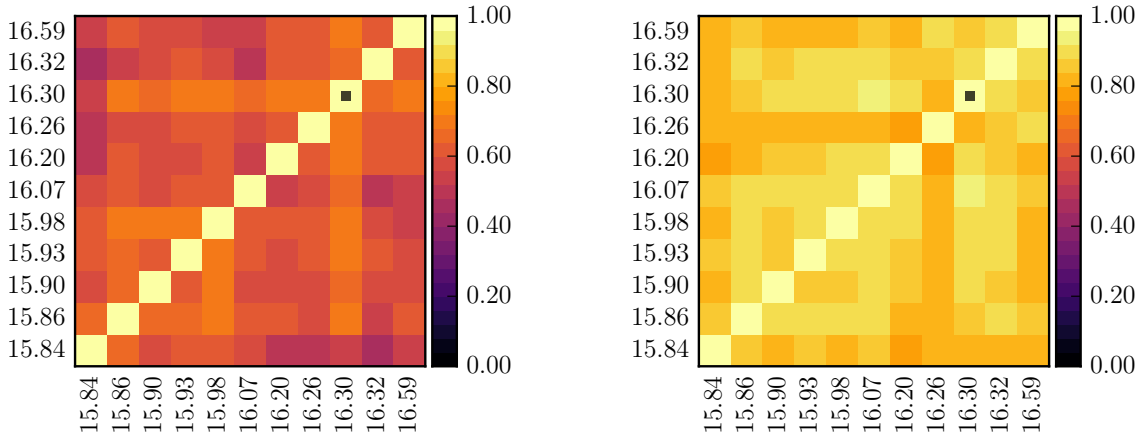
The absence of these features in the matrices means that the STARS data does not converge toward a particular spatial pattern, for example because the local dynamics in a particular region develop in a consistent way over the 500-year hindcast period, or during some part of it, e.g. from C_8 to C_{10} due to global warming. Furthermore, we do not find evidence that two chunks of the STARS output are more or less similar depending on their distance in time.

These observations bring into question the validity of using STARS to simulate a very long time period: the spatial dynamics that are generated by the simulation do seem to change over time, i.e. the κ-values fluctuate. However, this variability is likely to be of a random nature, a conclusion that is supported by our regression analysis in the previous section. We do not find anything suggesting that actual physical – or at least consistent – climate trends are at work over the simulation period.

At this point it is important to remind the reader that the dataset we examine was generated by using STARS well outside the spatial and temporal scope in which it was initially designed, and it is the result of one particular approach to downscaling the Pages2K data to a high resolution over a large territory. Our conclusions should thus be interpreted in this context only. It does not necessarily follow that STARS is unsuitable for smaller-scale simulations, although some arguments in that direction have surfaced in the past (see [7]).

4.2.2. Analysis by chunk mean temperature

In Fig. 4.11 we can see the same data as in the previous section, but this time ordered differently: moving from left to right, and from bottom to top, $\langle T_l(\mathcal{S})\rangle$ is increasing. In this representation, higher values close to the diagonal would imply that chunks more similar in their mean temperature are in higher spatial agreement. Monotonically increasing or decreasing rows would indicate that spatial patterns grow more similar the closer the mean temperature of the data is.

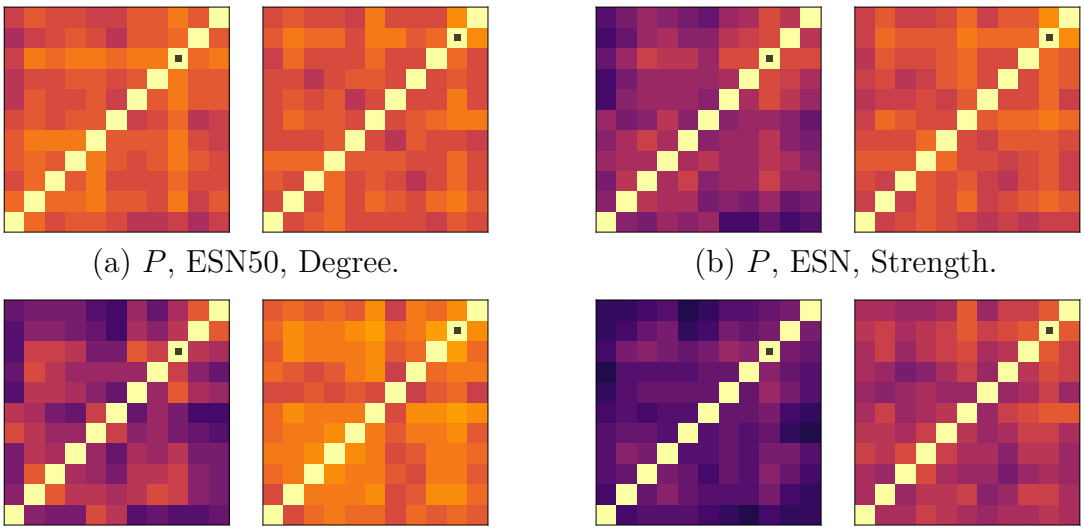


(a) P, Summer, ESN50, Degree.

(b) T_{mean}, Summer, TCN50, Degree.

Figure 4.11.: Spatial agreement between all chunks, ordered by mean seasonal temperature. Each label is the respective $\langle T_l(\mathcal{S})\rangle$ given in $°C$. The basis chunk B is marked by a black rectangle.

We find at most very faint clustering around the diagonal in either matrix, and in the matrices pertaining to the other networks and network measures, as can be seen in Figs. 4.12 and C.4. Where such features occur, for example in the center of Fig. 4.11b, they are not consistent across different climate variables or network measures, and thus more likely to be of random origin.



(a) P, ESN50, Degree.

(b) P, ESN, Strength.

(c) P, ESN, Directionality φ.

(d) P, ESN, Directionality ρ.

Figure 4.12.: Precipitation, ordered by $\langle T_l(\mathcal{S})\rangle$. The row and column containing the basis data are marked by a dark rectangle. The left matrix in each subplot is summer, and the right matrix winter data.

We conclude that the STARS mechanism does not produce more similar spatial patterns when the mean temperatures are close. Indeed, spatial variability seems to be largely unaffected by the mean temperature, with the same value range in all parts of a matrix given a particular combination of climate variable, season, network type, and network measure.

5. Conclusion

An important weak point of the STARS model is the limited spatial variability of its output. The resampling process is conditioned only at a very small number of geographic locations (in our case only one), and the date-to-date mapping that is generated there is simply extended to the whole simulation domain. In other words, STARS always resamples 'slices' of data covering all nodes, leaving little room for locally differentiated developments. Our hypothesis was that climatically relevant phenomena such as the Little Ice Age or industrialization – long-term trends that have substantially influenced the global and local climates during the simulation period – should be detectable in the data because of changes in their spatial variation.

We used climate networks to encode the spatio-temporal dynamics in a 500-year hindcast by STARS that was forced by reconstructed temperature data from the Pages2K project. We then computed three different network node measures over these networks. If the STARS dataset actually featured locally differentiated developments and long-term climate trends, we expect them to influence the network measure scores of our nodes over time. Finally, we used κ-statistics to detect if such changes occurred in the data in a consistent manner.

Our first finding is that STARS generates data that generally reproduces the spatial network patterns of its input data. Typical values of spatial agreement as measured by κ lie between 0.4 and 0.9, with the average value mostly depending on the combination of climate variable and network measure, and the fluctuation for a given combination generally about 0.2. Compared to the randomized data generated by our surrogates, which are consistently below 0.2 and often around 0.0, these results are highly significant. STARS thus can be said to keep the overall spatial structure of correlations in the original dataset intact, which remains true throughout all chunks considered. As a side note, the overall ranges of our κ-values are compatible with those found by Kücken et al. [35] who used cluster analysis instead of our network methodology to generate categorical maps. Working with 25 years worth of data instead of our 500 years, they analyzed a STARS-generated dataset structurally similar to ours that covered the territory of Germany. In their study, the agreement of STARS with observation data as measured by κ was found to be around 0.69.

The absolute values of our κ-statistic have to be put into perspective by comparison with our methodology's susceptibility to noise and general sensitivity. We are generally unable to distinguish STARS-generated data from those generated by adding random noise to the basis data, provided that the random noise does not exceed around 5% of the time series mean. The methodology is sufficient, however, to clearly differentiate between completely random data, randomized time series data that keeps location (our surrogates), and STARS data.

While we were able to use our surrogates to confirm the significance of our results insofar that there is no substantial baseline κ simply due to the 'climate data-like structure' of our datasets, we were unable to make more use of them. Specifically, a consistent use of z-scores over κ-scores in the analysis was unfortunately not possible. The main reason for this is that the surrogates were generally too different from the basis data, resulting in very low scores with a very narrow distribution. If our study is repeated, either the surrogates are generated with additional physically-motivated constraints on the reshuffling, or we could follow the example of Feldhoff et al. and bootstrap with whole seasons as blocks [8]. Another interesting approach used by Feldhoff et al. worth exploring in the future is based on randomized networks, similar to the ones we use to correct for spatial biases in the calculation of our strength and directionality measures, to establish points of reference.

Returning to our findings, we could not identify a clear trend of increasing or decreasing κ-scores between STARS chunks and the basis data over time, or with mean chunk temperature. Where such trends were visually indicated, a linear regression showed that their magnitude would be very small, with generally low r^2-values. We found one (weakly) significant regression for precipitation data using node strength as our network measure and ordering chunks by their mean temperature. Given that we could not confirm these results with any other combination of climate variable and network measure, however, we believe that they are not robust enough to warrant a reversal of our conclusions so far. We calculated $\kappa_{l,m}$ between all chunks, but found no substantial trends visible inside the STARS dataset, neither when ordered by time nor by chunk mean temperature.

As far as our methodology can show, this means that a STARS simulation does not become more similar to its input dataset as the mean temperatures of simulation period and observation period grow closer, or as the periods themselves come closer in time, nor are there indications of such trends within subsets of the STARS data, for example around the chunks that are associated with the Little Ice Age (C_2 through C_4) or industrialization (C_8 through C_{10}). In other words, since we expect our network measures to encode the statistical relationships between weather stations, we find no evidence that there are consistent long-term trends at play inside the STARS dataset that would lead to changes in spatial variability.

These findings might indicate that the simulation we analyzed, covering the territory of Germany over a period of 500 years, actually employed the STARS model outside the maximum spatial and temporal extent in which it can be considered appropriate. This somewhat reinforces the points made by Wechsung and Wechsung [7]. As found with our methodology, STARS does not seem to generate long-term climate trends in the simulation data that are rooted in a physical development of the system, e.g. by slowly evolving node relationships in our networks. Climatic phenomena like the Little Ice Age or the global warming associated with industrialization are not visible in the spatial variability of the dataset.

A few additional remarks are warranted when interpreting our results. First, it should be noted that Boris Orlowsky, the main author of STARS, explicitly discussed some of its limits, including the question of spatial and temporal scope, and offered some ideas on how to properly calibrate the model [1]. Our results reinforce

the notion that such preliminary work is highly important. Second, the dataset S that we analyzed was generated using a particular approach that introduced its own set of problems, including discontinuities and the circumvention of some of STARS' heuristic rules, as described in Sec. 2.2.1. On the other hand, the utilization of several short trend lines instead of one unified line covering all 500 years arguably should introduce more variability into the hindcast. Third, although Orlowsky offered a technique for multi-station simulations – in use today at PIK and other institutions – our data was generated using single-station STARS simulations, compounding the problems that arise from having a low number of representative stations used for resampling. So, in a sense, our dataset did not reflect the best possible STARS configuration to create spatial variability.

Of course, this study should only be considered a first exploration of the dataset. Next to better surrogates, allowing a more nuanced analysis of κ-score significance, it could be interesting to use a different set of network measures. Locally differentiated developments in the data could, for example, be more easily seen with network measures that are 'local' in the sense that they only consider the geographic neighbors of each node, or that they suppress long-range links. Alternatively, the analysis could be carried out with the Hamming distance metric used in [8], or the degree-conditional neighborhood dissimilarity metric developed in [6]. Another promising approach would be the comparison with output from other climate models, for example dynamical models that feature explicit parametrization for global warming, or between observational datasets that cover different periods, as they become available.

Our methodology of examining spatial variability using climate networks and κ-statistics can be applied to other climate models as well. It relies on reasonably long time series in order to compute meaningful similarity scores such as r or e, and on a sufficient number of nodes for network measure calculation. These requirements are, however, also a strength of the method: it allows the reduction of potentially very large datasets to relatively few key values that can be examined more closely. As the theory of climate networks continues to be developed by the scientific community both from a mathematical and physical point of view, the field becomes an increasingly useful approach to analyze the spatio-temporal dynamics present in climate data.

Bibliography

[1] B. Orlowsky, "Setzkasten Vergangenheit - ein Kombinatorischer Ansatz für Regionale Klimasimulationen", PhD thesis, Universität Hamburg, 2007.

[2] B. Orlowsky, F.-W. Gerstengarbe, and P. Werner, "A resampling scheme for regional climate simulations and its performance compared to a dynamical RCM", *Theoretical and Applied Climatology*, vol. 92, no. 3-4, pp. 209–223, 2008.

[3] M. Ahmed, K. Anchukaitis, B. M. Buckley, M. Braida, H. P. Borgaonkar, A. Asrat, E. R. Cook, U. Büntgen, B. M. Chase, D. A. Christie, *et al.*, "Continental-scale temperature variability during the past two millennia: Supplementary information", *Nature Geoscience*, vol. 6, no. 5, 2013.

[4] A. E. J. Ogilvie, "Little ice age", in *Encyclopedia of Global Warming & Climate Change*, S. Philander, Ed., vol. 3, SAGE Publications, 2012, pp. 864–865. DOI: `10.4135/9781452218564.n428`.

[5] T. Collins, "Industrialization", in *Encyclopedia of Global Warming & Climate Change*, S. Philander, Ed., vol. 3, SAGE Publications, 2012, pp. 768–771. DOI: `10.4135/9781452218564.n428`.

[6] S. Lange, J. F. Donges, J. Volkholz, and J. Kurths, "Local difference measures between complex networks for dynamical system model evaluation", *PLOS ONE*, vol. 10, no. 4, 2015.

[7] F. Wechsung and M. Wechsung, "Dryer years and brighter sky - the predictable simulation outcomes for Germany's warmer climate from the weather resampling model STARS", *International Journal of Climatology*, 2014. DOI: `10.1002/joc.4220`.

[8] J. H. Feldhoff, S. Lange, J. Volkholz, J. F. Donges, J. Kurths, and F.-W. Gerstengarbe, "Complex networks for climate model evaluation with application to statistical versus dynamical modeling of South American climate", *Climate Dynamics*, vol. 44, no. 5-6, pp. 1567–1581, 2014.

[9] A. Hagen-Zanker, "Measuring the performance of geosimulation models by map comparison", PhD thesis, Maastricht University, 2008.

[10] P. Werner and F. Gerstengarbe, "Proposal for the development of climate scenarios", *Climate Research*, vol. 8, no. 3, pp. 171–182, 1997.

[11] F.-W. Gerstengarbe, P. Werner, J. Lutz, P. Hoffmann, and C. Menz. (2015). STARS - STatistical Analogue Resampling Scheme, Potsdam Institute for Climate Impact Research, [Online]. Available: `https://www.pik-potsdam.de/research/climate-impacts-and-vulnerabilities/models/stars`.

[12] H. Lotze-Campen, L. Claussen, A. Dosch, S. Noleppa, J. Rock, J. Schuler, and G. Uckert, *Klimawandel und Kulturlandschaft Berlin*. Potsdam-Institut für Klimafolgenforschung, 2009.

[13] S. Huang, V. Krysanova, H. Österle, and F. F. Hattermann, "Simulation of spatiotemporal dynamics of water fluxes in germany under climate change", *Hydrological Processes*, vol. 24, no. 23, pp. 3289–3306, 2010. DOI: 10.1002/hyp.7753.

[14] S. Liersch, J. Cools, B. Kone, H. Koch, M. Diallo, J. Reinhardt, S. Fournet, V. Aich, and F. Hattermann, "Vulnerability of rice production in the inner niger delta to water resources management under climate variability and change", *Environmental Science & Policy*, vol. 34, pp. 18–33, 2013.

[15] W. Leal Filho, J. Lutz, J. Volkholz, and F.-W. Gerstengarbe, "Climate projections for southern africa using complementary methods", *International Journal of Climate Change Strategies and Management*, vol. 5, no. 2, pp. 130–151, 2013.

[16] F.-W. Gerstengarbe, P. C. Werner, H. Österle, and O. Burghoff, "Winter storm- and summer thunderstorm-related loss events with regard to climate change in germany", *Theoretical and Applied Climatology*, vol. 114, no. 3-4, pp. 715–724, 2013.

[17] P. Hoffman, H. Osterle, F.-W. Gerstengarbe, C. Holldorb, and F. Rumpel, "The effects of climate change on winter service in germany", *Routes/Roads*, no. 361, 2014.

[18] Potsdam-Institut für Klimafolgenforschung e.V. (Feb. 2016). GLOWA-Elbe: Auswirkungen des globalen Wandels auf Umwelt und Gesellschaft im Elbegebiet, [Online]. Available: https://www.pik-potsdam.de/glowa.

[19] Potsdam-Institut für Klimafolgenforschung e.V. and WetterOnline Meteorologische Dienstleistungen GmbH. (Feb. 2016). ClimateImpactsOnline, [Online]. Available: http://www.climateimpactsonline.com.

[20] H. Öesterle, F. Gerstengarbe, and P. Werner, "Die Elbe im globalen Wandel. Eine integrative Betrachtung.", in, F. Wechsung, V. Hartje, S. Kaden, M. Venohr, B. Hansjürgens, and P. Gräfe, Eds., ser. Konzepte für die nachhaltige Entwicklung einer Flusslandschaft. Weissensee-Verlag, 2014, vol. 9, ch. Ein meteorologischer Datensatz für Deutschland, 1951–2003, pp. 67–70. [Online]. Available: http://www.weissensee-verlag.de/autoren/Wechsung_213/9783899982138_kurz.pdf.

[21] A. Rheinwalt, N. Boers, N. Marwan, J. Kurths, P. Hoffmann, F.-W. Gerstengarbe, and P. Werner, "Non-linear time series analysis of precipitation events using regional climate networks for Germany", *Climate Dynamics*, 2015. DOI: 10.1007/s00382-015-2632-z.

[22] A. A. Tsonis and P. J. Roebber, "The architecture of the climate network", *Physica A*, no. 333, pp. 497–504, 2004. DOI: 10.1016/j.physa.2003.10.045.

[23] N. Malik, B. Bookhagen, N. Marwan, and J. Kurths, "Analysis of spatial and temporal extreme monsoonal rainfall over South Asia using complex networks", *Climate Dynamics*, vol. 39, no. 3-4, pp. 971–987, 2012.

[24] N. Boers, B. Bookhagen, N. Marwan, J. Kurths, and J. Marengo, "Complex networks identify spatial patterns of extreme rainfall events of the South American monsoon system", *Geophysical Research Letters*, vol. 40, no. 16, pp. 4386–4392, 2013.

[25] K. Yamasaki, A. Gozolchiani, and S. Havlin, "Climate networks around the globe are significantly affected by el nino", *Physical review letters*, vol. 100, no. 22, p. 228 501, 2008.

[26] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks: Structure and dynamics", *Physics Reports*, vol. 424, no. 4, pp. 175–308, 2006.

[27] J. Heitzig, J. F. Donges, Y. Zou, N. Marwan, and J. Kurths, "Node-weighted measures for complex networks with spatially embedded and sampled and or differently sized nodes", *The European Physical Journal B*, vol. 85, no. 1, pp. 1–22, 2012.

[28] L. Sachs and J. Hedderich, *Angewandte Statistik*, 12th ed. Springer-Verlag, 2006.

[29] R. Q. Quiroga, T. Kreuz, and P. Grassberger, "Event synchronization: A simple and fast method to measure synchronicity and time delay patterns", *Physical Review E*, vol. 66, 2002. DOI: `0.1103/PhysRevE.66.041904`.

[30] A. A. Tsonis, G. Wang, K. L. Swanson, and L. C. Rodrigues, "Community structure and dynamics in climate networks", *Climate Dynamics*, vol. 37, pp. 933–940, 2011.

[31] A. Rheinwalt, "Rainfall networks: Precipitation analysis for the region of Germany using complex networks", PhD thesis, Diploma thesis. Humboldt University, Berlin, 2011.

[32] E. W. Weisstein, "Voronoi diagram", *MathWorld - A Wolfram Web Resource*, 1999. [Online]. Available: `http://mathworld.wolfram.com/VoronoiDiagram.html`.

[33] A. Rheinwalt, N. Marwan, J. Kurths, P. Werner, and F.-W. Gerstengarbe, "Boundary effects in network measures of spatially embedded networks", *EPL (Europhysics Letters)*, vol. 100, no. 2, p. 28 002, 2012.

[34] N. Boers, A. Rheinwalt, B. Bookhagen, H. M. Barbosa, N. Marwan, J. Marengo, and J. Kurths, "The South American rainfall dipole: A complex network analysis of extreme events", *Geophysical Research Letters*, vol. 41, no. 20, pp. 7397–7405, 2014.

[35] M. Kücken, F.-W. Gerstengarbe, and B. Orlowsky, "A combination of cluster analysis and kappa statistic for the evaluation of climate model results", *Journal of Applied Meteorology and Climatology*, vol. 48, no. 9, pp. 1757–1765, 2009.

[36] J. Cohen, "A coefficient of agreement for nominal scales", *Educational and Psychological Measurement*, vol. 20, no. 1, 1960.

[37] R. G. Pontius, "Quantification error versus location error in comparison of categorical maps", *Photogrammetric Engineering and Remote Sensing*, vol. 66, no. 8, pp. 1011–1016, 2000.

[38] B. Orlowsky and J. Lutz, *User guide to the STatistical Analogue Resampling Scheme - version 2.4*, Potsdam Institute for Climate Impact Research, 2013.

[39] J. Cohen, "Weighted kappa: Nominal scale agreement with provision for scaled disagreement or partial credit", *Psychological Bulletin*, vol. 70, no. 4, 1968.

[40] J. H. Fleiss and J. Cohen, "The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability", *Educational and Psychological Measurement*, vol. 33, 1973.

[41] A. Hagen, "Fuzzy set approach to assessing similarity of categorical maps", *International Journal of Geographical Information Science*, vol. 17, no. 3, pp. 235–249, 2003.

[42] J. F. Donges, J. Heitzig, B. Beronov, M. Wiedermann, J. Runge, Q. Y. Feng, L. Tupikina, V. Stolbova, R. V. Donner, and N. Marwan, "Unified functional network and nonlinear time series analysis for complex systems science: The pyunicorn package", *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 25, no. 11, p. 113 101, 2015.

Appendices

A. Climate model and data

A.1. STARS model

General scheme

Let $X = \{x, y, \dots\}$ be a tuple of climate variables such as mean temperature, total precipitation or cloud cover. We then call

$$^{R}X_i(t_0) = \left\{^{R}x_i(t_0), {}^{R}y_i(t_0), \dots\right\} \tag{A.1}$$

an *observation* or *tuple* in the dataset R located at some point i with the geographic location $(\varphi, \lambda)_i$ and some time t_0 in the period $^{R}t_{\min} \leq t_0 \leq {}^{R}t_{\max}$. Consequently, we denote with $^{R}X_i(t)$ the time series of ^{R}X, and with $^{R}x_i(t)$ the time series of ^{R}x at point i.

STARS resamples the observed (real) time series of tuples $^{a}X_i(t)$ to construct a simulated time series $^{S}x_i(t)$ for the simulation period $^{S}t_{\min} \leq t \leq {}^{S}t_{\max}$, located at the same coordinates. It does so by defining a date-to-date mapping

$$^{S}t = {}^{S}(d, m, y) \leftarrow {}^{R}t = {}^{R}(d, m, y) \tag{A.2}$$

where d, m and y are the day, month and year. In other words, the model is a surjective function $f(^{S}t) = {}^{R}t$ that assigns to each date in the simulation period one date from the observation period.

At the individual location i, the model assembles $^{S}X_i(t)$ by resampling first whole years and then blocks of length l taken from $^{R}X_i(t)$, so that $^{S}X_i(t_1) \leftarrow {}^{R}X_i(t_2)$ for all times $t_1 \in \{^{S}t_0, \dots, {}^{S}t_{\max}\}$, and some $t_2 \in \{^{R}t_0, \dots, {}^{R}t_{\max}\}$. This is based on the assumption that 'weather states from segments of the observation period may occur again or very similarly during the simulation period' [2, p. 221].

Predictor and heuristics

In the most general way, STARS uses the slope a and intercept b of a linear regression performed on the *forcing variable*, or *predictor*, ω that captures the 'essential climate variability in the region of interest' [2, p. 221]. The regression line

$$\omega = a \cdot t + b \tag{A.3}$$

is also called the *trend line* and used to condition the resampling of $^{R}X_i(t)$ into $^{S}X_i(t)$. The aim is to obtain a simulated series so that

$$^{S}a = a \pm \epsilon \quad \text{and} \quad {}^{S}b = b \pm \epsilon \tag{A.4}$$

with some small tolerance ϵ and ^{S}a and ^{S}b the parameters of a regression performed on the simulated series' ω. At the same time, STARS ensures physical properties of the time series through some additional conditions:

i **Annual cycles** The time series ${}^{S}X_i(t)$ should exhibit realistic annual cycles: lower temperatures in the winter months, more rain in the summer, for example.

ii **Persistence** The time series ${}^{S}X_i(t)$ should feature realistic persistence: no single cold days in an otherwise very warm period.

iii **Tuple consistency** Each tuple ${}^{S}X_i$ has to be consistent: a simulation should not generate dates without cloud coverage but with heavy rainfall.

Conditions (i) and (ii), as well as Eq. (A.4), restrict the resampling process. The condition (iii) is met since STARS only resamples whole tuples, so every ${}^{S}X_i$ generated has precedence in reality.

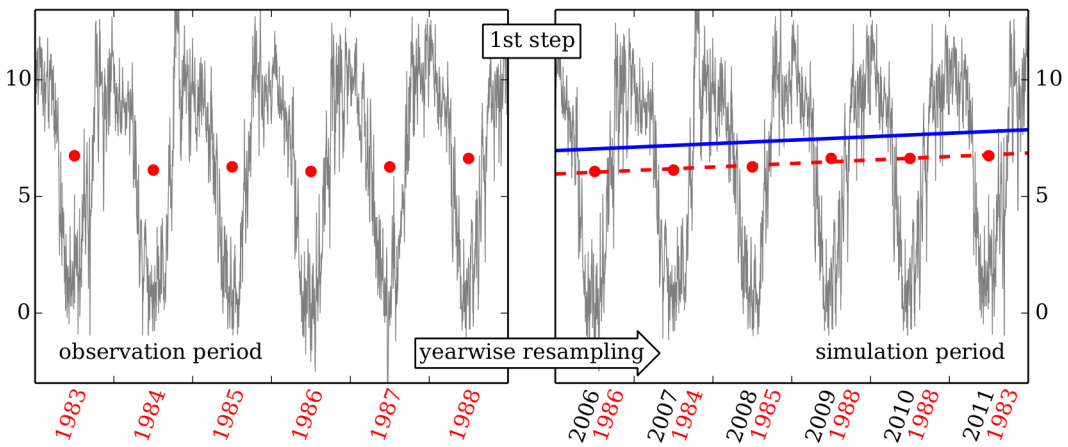
Resampling process

In a first approximation, STARS resamples the original observation period year-wise, generating an ensemble of year-to-year mappings $\left({}^{S}(y) \leftarrow {}^{R}(y)\right)_k$, where y denotes the year and typically $k = 1, \ldots, 1000$ (Fig. A.1a). It then choses the year-mapping that best satisfies Eq. (A.4) and refines it (Fig. A.1b). This second step is done iteratively with exaggerated parameters.

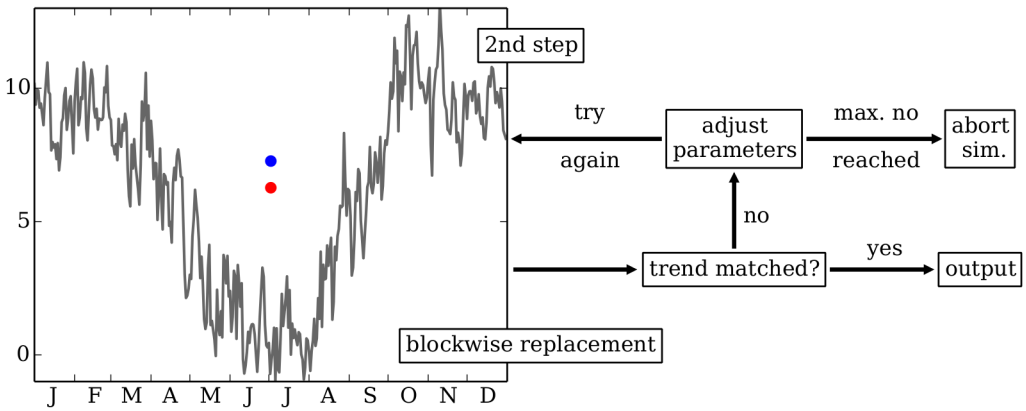
Let $\left({}^{S}(y) \leftarrow {}^{R}(y)\right)_m$ be the mapping chosen in the first step, resulting in the preliminary simulated series ${}^{m}X_i(t)$. STARS then analyzes ${}^{m}X_i(t)$ by blocks of length l, classifying each block using cluster analysis. It calculates the contribution of the individual blocks to the overall inequality in Eq. (A.4) and marks the largest contributers for replacement.

Block replacement is conditioned on a number of heuristic rules that are designed to enforce conditions (ii) and (iii) while improving the equality in Eq. (A.4). First, blocks are only replaced by blocks from the same cluster as determined by the cluster analysis. Second, only blocks not yet sampled are considered. Third, the calendar distance between a block and its replacement should not exceed some threshold. Last, the candidate pool only contains blocks that fit with the predecessor and successor blocks.

If the refined simulation does not yet satisfy Eq. (A.4), the second step is repeated with exaggerated parameters, leading to more blocks in ${}^{m}X_i(t)$ being marked for replacement.



(a) Year-wise sampling. The left window shows the original observation period with yearly means (red dots). The right window features the simulated period with yearly means (red dots), the prescribed (blue line) and current regression line (red line).



(b) Block-wise sampling. Replacement of 12-day-blocks to better approach the prescribed yearly mean (blue dot) with the yearly mean of the current simulation series (red dot).

Figure A.1.: Illustration of the two resampling levels found in the STARS model as shown on [11]. The forcing variable, or predictor, ω in this example is the mean temperature.

Multi-location simulation

When STARS is extended to multiple locations, Eq. (A.4) and the conditions (ii) and (iii) are to be met there as well. This is formulated as another condition:

iv **Spatial consistency** A simulated field ${}^S\vec{X}(t) = \left({}^SX_1(t), \ldots, {}^SX_N(t)\right)$ has to be consistent: we do not expect snow and hot temperatures concurrently at adjacent stations.

STARS meets (iv) since if more than one geographic location is simulated, the mapping is identical across locations. Thus, the spatial field ${}^S\vec{X}(t)$ of the climate variable tuple ${}^SX$ at all N locations for some point in time t remains realistic since it is an exact copy of an observed field, ${}^S\vec{X}(t_1) = {}^R\vec{X}(t_2)$, $t_1 \in \{{}^St_0, \ldots, {}^St_{\max}\}$ and $t_2 \in \{{}^Rt_0, \ldots, {}^Rt_{\max}\}$.

STARS reduces the dimensionality of the problem by using cluster analysis to find a 'small set of stations [...] which represents the variability among the stations in a generalising way.' Separate regression parameters are then prescribed for each of the $M \ll N$ representative stations, allowing 'for the simulation of spatially differentiated developments' [2].

Note that in this study, we examine a dataset that was generated using the single-station operating mode of STARS.

STARS example output

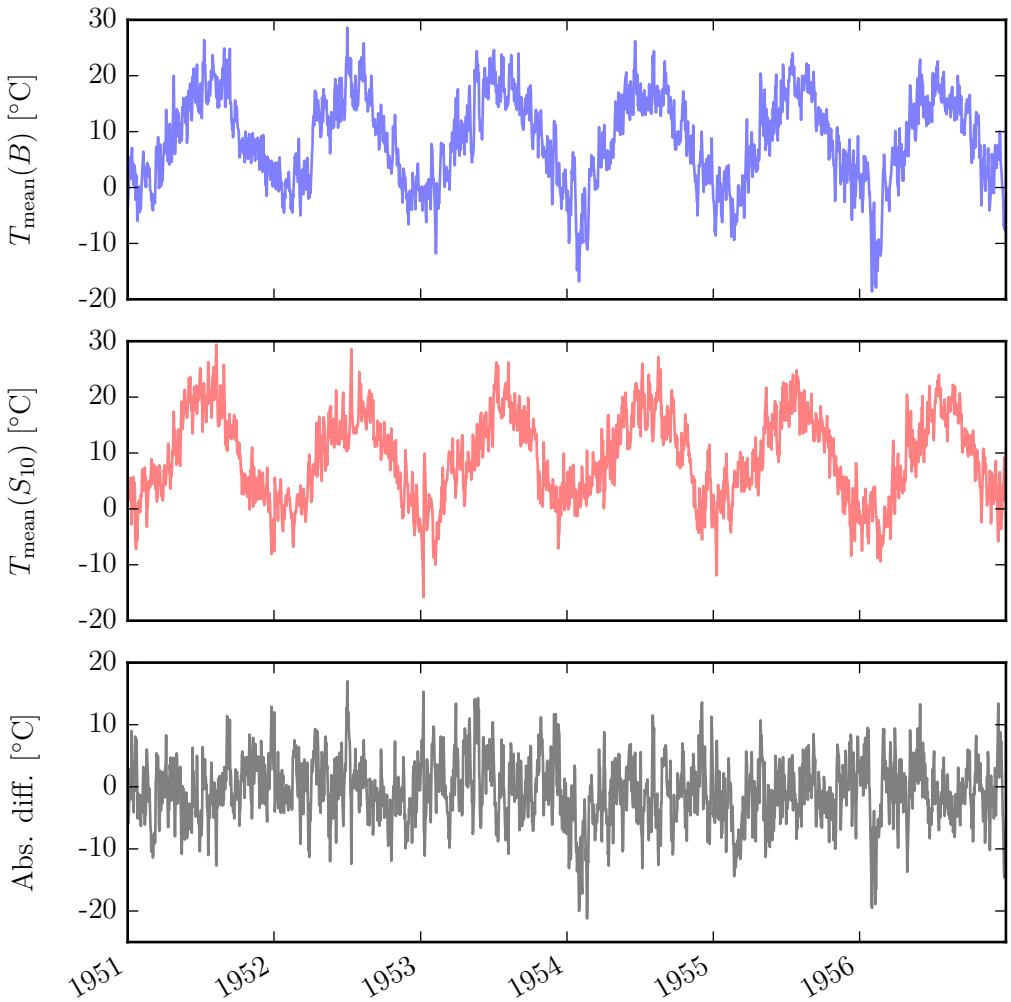


Figure A.2.: Example of a STARS time series and comparison with the input data, using mean daily temperature time series at the weather station in Potsdam between 1 January 1951 and 31 December 1956.

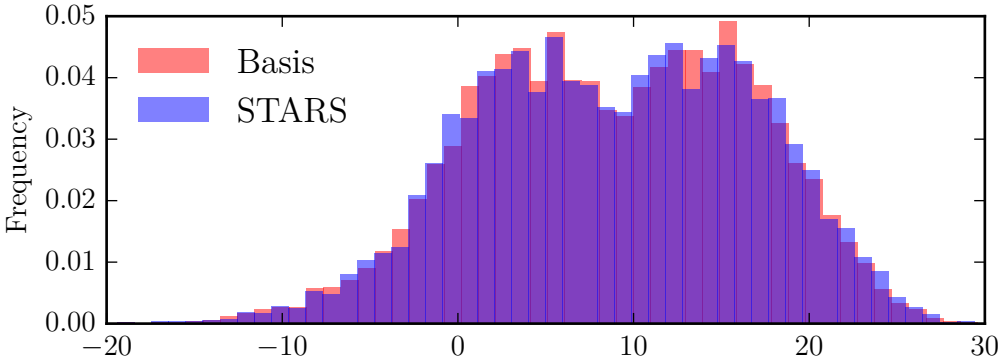


Figure A.3.: Comparison of frequency distributions in the STARS and basis data. We show the mean daily temperature time series at the weather station in Potsdam between 1 January 1951 and 31 December 1999.

A.2. Parameters and forcing used to generate S

The parameters in Tab. A.1 were used by Jan Volkholz to generate the simulation dataset S, an illustration of his method that divides the simulation period into five-year chunks can be seen in Fig. A.4.

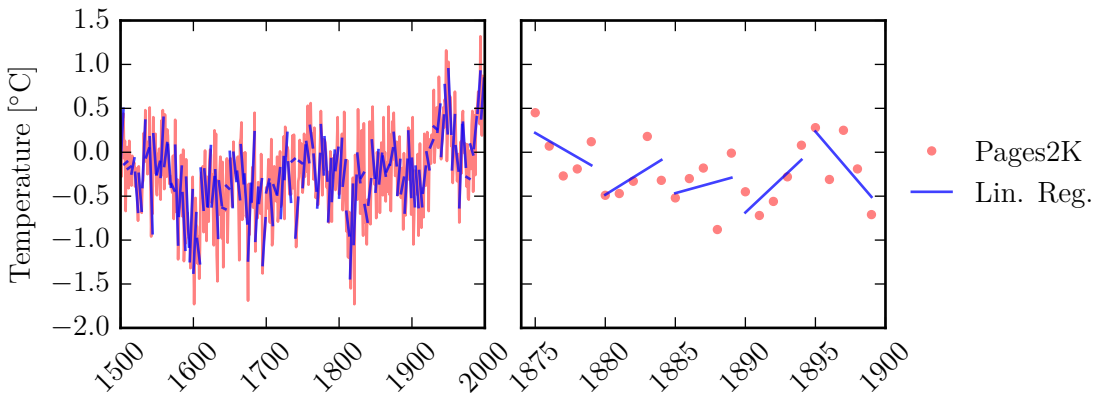


Figure A.4.: Creation of trend lines by Jan Volkholz. Displayed are the temperature anomalies of the Pages2K Europe datasets, and the linear regression trend lines created from 5-year periods of these values. The right plot is a closeup.

Table A.1.: Parameter file used in the STARS run. The different parameters are explained in detail in [38].

```
normalise=y
internal_cluster=y
no_clusters=40
output_block=n
no_sims=1
char_var=5
blocklength=12
basz_path=[...]

stats=27008
calc_trends=n
delta_temp=3.0
tolerances=0.2
tol_jd=20

no_conca=50000
```

B. Methodology

B.1. Mathematical notes

Sample standard deviation and covariance

The sample standard deviation σ_{x_i} and the sample covariance $\mathrm{cov}(x_i, x_j)$ of two time series $x_i(t) = \{x_i(t_0), \ldots, x_i(t_n)\}$ and $x_j(t) = \{x_j(t_0), \ldots, x_j(t_n)\}$ are defined as

$$\mathrm{cov}(x_i, x_j) = \frac{\sum_{t=t_0}^{t_n} (x_i(t) - \langle x_i \rangle)(x_j(t) - \langle x_j \rangle)}{n-1}, \tag{B.1}$$

$$\sigma_{x_i} = \sqrt{\frac{\sum_{t=t_0}^{t_n} (x_i(t) - \langle x_i \rangle)^2}{n-1}}, \tag{B.2}$$

with $\langle x_i \rangle = {}^1\!/_n \sum_t x_i(t)$ denoting the sample mean, and n the length of the time series.

Hypergeometric distribution

To find the expected number of synchronization counts, we calculate the probability mean of the hypergeometric distribution in Eq. (3.6) as

$$\langle c \rangle_p = \sum_{k=0}^{c} p(k)\ k, \tag{B.3}$$

and its standard deviation as

$$\sigma_p = \sqrt{\sum_{k=0}^{c} p(k)\ (k - \langle c \rangle_p)^2}. \tag{B.4}$$

Node-splitting invariance

For sufficiently dense sampling, the introduction of additional stations inside the domain would yield almost the exact same observations and would thus create redundant nodes.

Heitzig et al. point out that the creation of a redundant node is mathematically equivalent to splitting a network node k in two, creating the nodes k' and k'', each covering half the area of the original domain. The new nodes should display a similar if not identical pattern of links to other nodes and are thus redundant. We define $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ as the original graph, and $\mathcal{G}' = (\mathcal{N}', \mathcal{L}')$ as the new graph containing the two nodes k', k'' instead of k. It follows that

$$\mathcal{N}' = \mathcal{N} \backslash \{k\} \cup \{k', k''\},$$
$$\mathcal{L}' = \{\{i,j\} \in \mathcal{L} : i, j \neq k\} \cup \{\{i,k'\}, \{i,k''\} : \{i,k\} \in \mathcal{L}\} \cup \{\{k', k''\}\},$$

since k' and k'' are linked to the same nodes as k was, and are linked between each other. If the area covered by k in $\mathcal{G}$ is expressed as a node weight v_k, the new weights satisfy $v_{k'} + v_{k''} = v_k$.

Generally speaking, a network measure f that assigns a score f_k to node k is not invariant to such splitting, i.e. the two equations

$$f_k(\mathcal{G}) = f_k(\mathcal{G}') \tag{B.5}$$

$$f_{k'}(\mathcal{G}') = f_{k''}(\mathcal{G}') = f_k(\mathcal{G}) \tag{B.6}$$

are not satisfied. The average node degree, for example, increases if redundant nodes are introduced. In an effort to reduce the impact of the domain discretization on network measures, Heitzig et al. introduced 'node splitting invariance', or n.s.i. for short [27]. Their proof that Eq. (3.10) is indeed invariant to node splitting, i.e. it satisfies Eqs. (B.5) and (B.6) is as follows:

$$\mathcal{N}_{k'}^{+}(G') = \mathcal{N}_{k''}^{+}(G') = \mathcal{N}_{k}^{+}(G)\backslash\{k\} \cup \{k', k''\},$$
$$\mathcal{N}_{i}^{+}(G') = \mathcal{N}_{i}^{+}(G)\backslash\{k\} \cup \{k', k''\},$$

and $\mathcal{N}_{j}^{+}(G') = \mathcal{N}_{j}^{+}(G)$ for all $i \in \mathcal{N}_{k}^{+}(G)$ and $j \in \mathcal{N}\backslash\mathcal{N}_{k}^{+}(G)$. For more details see the cited paper.

B.2. Categorization of network measures

The categorization of a sample of discrete values follows the scheme detailed in this section. To find the categories for some scalar field $\vec{k}$ we consider the set of all its values $\{k_1, ..., k_N\}$. We arrange the set in increasing order of magnitude for the so-called order statistics

$$\{k_i,\ i = 1, 2, \ldots, N : k_{i+1} \geq k_i,\ \forall\, i = 1, 2, \ldots, N-1\}, \tag{B.7}$$

a sorted vector in which the rank of each value v is equal its position i. We can estimate a percentile value by finding the ranks directly above and below the true value and linearly interpolate between them: to find the percentile $P \in [0, 100]$, we define a linear interpolation function $k(x)$ that satisfies $\{k(x),\ x \in [1, N] : k(i) = k_i,\ \forall\, i = 1, \ldots, N\}$:

$$x(P, N) = \frac{P}{100}(N-1) + 1, \tag{B.8}$$

$$k(x) = k_{\lfloor x \rfloor}\Big(x - \lceil x \rceil\Big) + k_{\lceil x \rceil}\Big(x - \lfloor x \rfloor\Big), \tag{B.9}$$

using the floor and ceiling functions $\lfloor x \rfloor = \max\{m \in \mathbb{Z} : m \leq x\}$ and $\lceil x \rceil = \min\{n \in \mathbb{Z} : n \geq x\}$. Consequently, a node i is placed in category l with regard to field $\vec{k}$, i.e. $k'_i = l$, iff $v(x(P_l, N)) \leq k_i < v(x(P_{l+1}, N))$.

Fig. B.1 shows the distributions of a sample $\vec{k}$ and of the corresponding $\vec{k}'$, and Fig. B.2 illustrates the categorization of a sample vector $\vec{k}$.

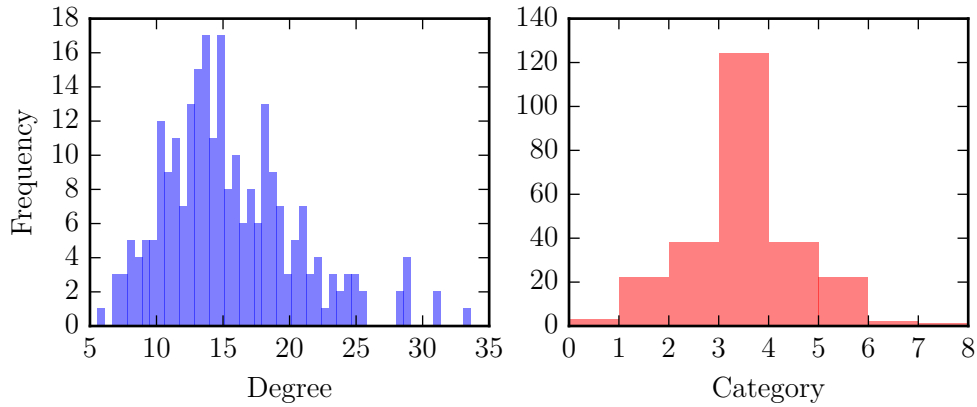


Figure B.1.: Distribution of degree scores before and after the categorization based on percentiles.

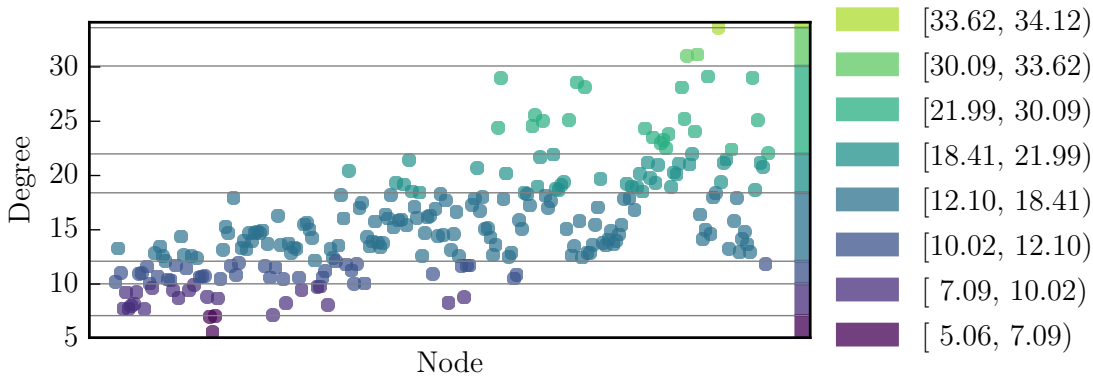


Figure B.2.: Conversion of $\vec{k}$ to $\vec{k}'$ for 250 nodes. The category boundaries based on Eq. (3.20) are listed on the right.

B.3. κ-statistics

κ-statistics for categorical maps

Pontius proposed to complement Cohen's κ with κ_{loc} and κ_{quan}, designed to judge the agreement of two categorical maps by their agreement with regards to the locations of categories, and the sizes of categories, respectively. Hagen-Zanker later discovered problematic behavior when numerically calculating κ_{quan}, including instability, asymmetry and out-of-range values [9]. These problems could be reproduced for this study, and we thus also implemented the alternative κ_{his}, proposed by Hagen-Zanker, instead of Pontius' original κ_{quan} [37]. The three statistical measures are thus

$$\kappa = \frac{p_O - p_C}{1 - p_C}, \qquad \kappa_{\text{loc}} = \frac{p_O - p_C}{p_M - p_C}, \qquad \kappa_{\text{his}} = \frac{p_M - p_C}{1 - p_C} \tag{B.10}$$

with the same notation as described in Eq. (3.23) and (3.25). These definitions have the added benefit that $\kappa = \kappa_{\text{loc}} \cdot \kappa_{\text{his}}$.

Pontius suggested the following scheme to guide interpretation of his κ-values. Consider the case of a real categorical map R, and its simulated counterpart S, both containing data classified into c categories. The simulation's quantification error is high if the quantity of a particular category differs strongly between S and R. Similarly, the simulation's location error is high if the spatial distribution of the cells belonging to a category differs strongly between S and R. Pontius classifies a spatially explicit simulation 'according to its ability to specify both quantity and location' of categories from no ability (NQ/NL), medium ability (MQ/ML), to perfect ability (PQ/PL).

The proportion of cells in category l is $p_{l\cdot}$ on the simulated map and $p_{\cdot l}$ on the real map. Pontius derives the expected proportions of chance agreement for simulations with differing abilities to specify quantity and location. The results are displayed in Tab. B.1 and constitute the basis for the proposed subdivision of κ into κ_{loc} and κ_{his}.

Table B.1.: Expected proportion of chance agreement with reality given a simulation's ability to correctly specify location and quantity.

	Location		
Quantity	No (NL)	Medium (ML)	Perfect (PL)
No (NQ)	$1/c$	$1/c + \kappa_{\text{loc}}(\text{NQPL} - 1/c)$	$\sum_{n=1}^{c} \min(1/c, p_{\cdot n})$
Medium (MQ)	$\sum_{m=1}^{c} p_{m\cdot} p_{\cdot m}$	p_O	$\sum_{m=1}^{c} \min(p_{m\cdot}, p_{\cdot m})$
Perfect (PQ)	$\sum_{n=1}^{c} p_{\cdot n} p_{\cdot n}$	$\text{PQNL} + \kappa_{\text{loc}}(1 - \text{PQNL})$	1

We can list the information gained from each type of κ in Tab. B.2. It becomes obvious that the simple scheme of interpretation that was proposed by Pontius is not directly applicable to κ_{his}, which was designed to replace κ_{quan}.

Table B.2.: Different κ-types and the underlying assumption about the model to be tested, according to Pontius. κ_{his} was later added by Hagen-Zanker and is more difficult to interpret in the same scheme.

κ-Statistic	Formula	Information yield
κ	$\dfrac{p_O - \text{MQNL}}{PQPL - \text{MQNL}}$	Success attributable to simulation compared to MQNL, divided by maximum success possible by perfect simulation
κ_{no}	$\dfrac{p_O - \text{NQNL}}{PQPL - \text{NQNL}}$	Success attributable to simulation compared to NQNL, divided by maximum success possible by perfect simulation
κ_{loc}	$\dfrac{p_O - \text{MQNL}}{\text{MQPL} - \text{MQNL}}$	Success attributable to simulation due to location specification, divided by maximum possible success due to perfect location specification
κ_{quan}	$\dfrac{p_O - \text{NQML}}{\text{PQML} - \text{NQML}}$	Success attributable to simulation due to quantity specification, divided by maximum possible success due to perfect quantity specification
κ_{his}	$\dfrac{\text{MQPL} - \text{MQNL}}{PQPL - \text{MQNL}}$	

Consequently, even though κ_{his} has been used successfully in the past as an indicator

for agreement between categorical maps [31], [35], we decided that its exclusion from this study due to the nature of our categorization process was acceptable.

Weighted κ_w

Cohen himself suggested an extension of κ for situations in which disagreements between raters can be considered more or less severe. In the original κ, all discrepancies are considered equal, as is appropriate for nominal categories. Its weighted counterpart κ_w can be used to weigh a larger discrepancy more heavily, as is sensible if the categories are ordered [39], [40].

κ_w is calculated by assigning weights to the cells in the agreement matrix, i.e. scaling of the disagreement (or agreement, depending on the problem). In the following we will consider disagreement weights.

First, κ is expressed in terms of observed and chance disagreement,

$$\kappa = 1 - \frac{q_O}{q_C}, \tag{B.11}$$

where $q_O = 1 - p_O$ and $q_C = 1 - p_C$ are the observed and expected proportions of disagreement.

Second, q_O and q_C are replaced by their weighted counterparts. A disagreement weight $v_{mn} \geq 0$ is assigned to the cell (m, n) in the contingency table. Note that κ_w is invariant to positive multiplicative transformation of the weights. For convenience, the diagonal can be set to zero, and some $v_{\max}$ is chosen, with the rest of the ratio weights in the interval $[0, v_{\max}]$. In short,

$$q \to q^* = \frac{1}{v_{\max}} \sum_{m=1}^{c} \sum_{n=1}^{c} v_{mn}\, p_{mn} \tag{B.12}$$

for both q_O and q_C.

Third, the weighted coefficient is calculated as

$$\kappa_w = 1 - \frac{q_O^*}{q_C^*} = 1 - \frac{\sum_m \sum_n v_{mn}\, p_{O,mn}}{\sum_m \sum_n v_{mn}\, p_{C,mn}}. \tag{B.13}$$

Note that κ_w is still fully chance-corrected since both q_C^* and q_O^* are calculated using the same weights.

As indicated, κ_w can be used for categories that are ordered in some sense. In this study, categories constitute a set of total order, they fulfill the conditions of antisymmetry (if $k \leq m$ and $m \leq n$ then $k \leq n$), transitivity (if $m \leq n$ and $n \leq m$ then $n = m$), and totality ($n \leq m$ or $m \leq n$). In the contingency table, they were sorted in ascending order and weighted with quadratic disagreement weights:

$$v_{mn} = (m - n)^2. \tag{B.14}$$

The application of κ_w did not, however, noticeably influence the results. For reasons of clarity, we thus decided to proceed with the unweighted κ.

Implementation of κ in Python

We implemented the different κ-types in Python2.7, also including weighted κ and fuzzy κ-statistics as suggested by [41].

Listing B.1: Implementation of Cohen's κ and the additional κ_{his} and κ_{loc} with support for weights in Python 2.

```
import numpy as np

def categorize_data(data1, data2, global_buckets):
        """
        Categorizes observations in data1 and data2
        into some provided `global_buckets`
        (categories), or generates new buckets from
        percentile values over the combined `data1`
        and `data2` data.

        Arguments:
        ----------
        data1 : list or 1D array
            A list of values, e.g. a vector containing
            the degree of every node on a network.

        data2 : list or 1D array
            Another list of values that is to be
            compared to data1.

        global_buckets : array-like or False, optional
            The set of global buckets (or categories)
            to use, if False two sets will be
            calculated from quantiles specified in the
            global `BUCKET_PERCENTILES`.


        Returns:
        --------
        cat_data1 : list or 1D array
            A list of categorized data.

        cat_data2 : list or 1D array
            A list of categorized data.
        """
        # make sure we have arrays
        data1 = np.array(data1)
        data2 = np.array(data2)

        # either we were supplied the categories,
        # or we compute them from quantiles
        if type(global_buckets) == np.ndarray:
            buckets1 = buckets2 = global_buckets
        elif global_buckets is False:
            buckets1 = np.percentile(data1,
                BUCKET_PERCENTILES)
            buckets2 = np.percentile(data2,
                BUCKET_PERCENTILES)

        # categorizing the data
        cat_data1 = np.digitize(data1, buckets1)
        cat_data2 = np.digitize(data2, buckets2)
```

```
        return cat_data1, cat_data2


def compute_crosstab(cat_data1, cat_data2):
    """
    Computes the crosstab from categorized data.

    .. note::
    This is also known as a confusion matrix. An
    element :math:`C_{i, j}` is equal to the
    number of obserations that, according to one
    rater, belong to category :math:`i`, but
    were rated in category :math:`j` by the other.

    Arguments:
    ----------
    cat_data1 : list or 1D array
        A list of categorized data.

    cat_data2 : list or 1D array
        A list of categorized data.


    Returns:
    --------
    ct : 2D array
        The square crosstab (or contingency table)
        of the data provided.
    """
    # number of categories
    n_cats = max(max(cat_data1), max(cat_data2)) + 1

    # fill crosstab
    ct = np.zeros((n_cats, n_cats), dtype=np.int)

    data = np.vstack((cat_data1, cat_data2))

    for i in range(data.shape[1]):
        ct[data[0, i], data[1, i]] += 1

    return ct


def calculate_kappa(crosstab, weights):
    """
    Calculates Cohen's :math:`\kappa` as well
    as the complementary measures
    :math:`\kappa_{loc}` and :math:`\kappa_{his}`.


    Arguments:
    ----------
    crosstab : 2D array
        A square crosstab (or contingency table)
        of some data.

    weights : None, string or array
        If any and which weights shall be used.
```

```
        Available options:

        * None: no weights will be used.
        * 'linear': linear weights (Cohen 1968)
        * 'quadratic': quadratic weights (Fleiss
          and Cohen 1973)
        * 2D array of (n_cats * n_cats): custom
          weights


    Returns:
    --------
    out : tuple of three floats
        :math:`\kappa, \kappa_{loc}, \kappa_{his}`
    """
    if crosstab.shape[0] != crosstab.shape[1]:
        raise TypeError

    n_cats = len(crosstab)

    # construct weights array
    if isinstance(weights, str):
        wt_scheme = weights
        weights = None
    else:
        wt_scheme = ''
    if weights is None:
        weights = np.empty((n_cats, n_cats))
        for i in range(n_cats):
            for j in range(n_cats):
                diff = abs(i - j)
                if wt_scheme == 'linear':
                    weights[i, j] = diff
                elif wt_scheme == 'quadratic':
                    weights[i, j] = diff ** 2
                elif not wt_scheme:  # unweighted
                    weights[i, j] = bool(diff)
                else:
                    raise ValueError

    # sum of elements
    n = float(crosstab.sum())

    # observed proportion in agreement
    ob = crosstab / n
    po = 1.0 - np.sum(ob * weights)

    # margins
    row_sum = ob.sum(axis=1)
    col_sum = ob.sum(axis=0)

    if False:
        """
        Calculation as defined by Pontius 2000
        and Hagen 2003. Results in consistent
        :math:`\kappa, \kappa_{loc}`, but
        :math:`\kappa_{quan}` is often above one.
        """
        NQNL = 1 / float(n_cats)
```

```
        NQPL = np.sum(np.minimum(NQNL * np.ones(
            row_sum.shape), row_sum))
        MQNL = np.sum(row_sum * col_sum)

        kstd = (po - MQNL) / float(1 - MQNL)

        MQPL = np.sum(np.minimum(col_sum, row_sum))

        kquanHagen = (MQPL - MQNL) / float(1 - MQNL)

        PQNL = np.sum(row_sum ** 2)

        kloc = (po - MQNL) / float(MQPL - MQNL)

        NQML = NQNL + kloc * (NQPL - NQNL)
        PQML = PQNL + kloc * (1 - PQNL)

        kno = (po - MQNL) / float(1 - NQNL)
        kquanPontius = (po - NQML) / float(
            PQML - NQML)

        print 'kstd         %.8f' % kstd
        print 'kno          %.8f' % kno
        print 'kloc         %.8f' % kloc
        print 'kquanPontius %.8f' % kquanPontius
        print 'kquanHagen   %.8f' % kquanHagen

    # maximum possible proportion in agreement, given
    # marginal proportions
    po_max = np.sum(np.min(zip(row_sum, col_sum),
                    axis=1))

    # expected proportion in agreement, given chance
    ex = np.outer(row_sum, col_sum)
    pe = 1.0 - np.sum(ex * weights)

    # if all weights zero, nothing matters
    if np.count_nonzero(weights):
        kappa = (po - pe) / (1 - pe)
        kappa_location = (po - pe) / (po_max - pe)
        kappa_histogram = (po_max - pe) / (1 - pe)
    else:
        kappa = 1.0
        kappa_location = 1.0
        kappa_histogram = 1.0

    return kappa, kappa_location, kappa_histogram
```

B.4. Surrogates

Surrogate z-scoring

Our $\kappa(C_l^s, C_m)$-values are normally distributed, as can be seen in Fig. B.5 and B.6. If we can establish that the surrogate distributions are generally similar, we can use $z(C_l, C_m)^*$ instead of $z(C_l, C_m)$ without giving up too much information. This approach seems justified, since we are not so much interested in the absolute value

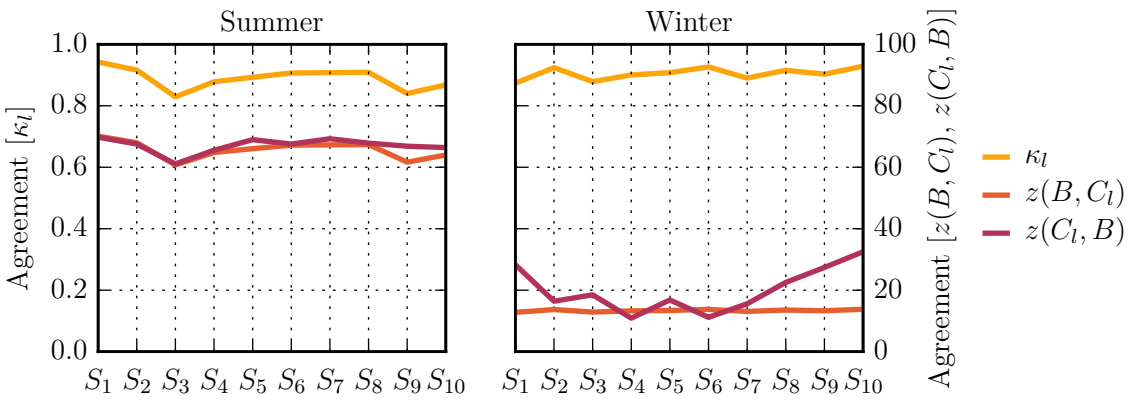
of z, but only want to test whether it is close to zero.

Normal distributions are fully characterized by their mean and standard deviation. Using $\kappa(C_l^s, B)$ as an example, we tested whether the mean and standard deviation of $\kappa(B^s, B)$ lie inside n standard deviations of the mean of the means and standard deviations for $\kappa(C_l^s, B)$ for all l. In other words, we examine whether the following equations are satisfied:

$$\langle \kappa(B^s, B) \rangle = \left(\frac{1}{10} \sum_l \langle \kappa(C_l^s, B) \rangle \right) \pm n \cdot \sigma(\langle \kappa(C_l^s, B) \rangle)$$
$$\sigma_{\langle \kappa(B^s,B) \rangle} = \left(\frac{1}{10} \sum_l \sigma_{\kappa(C_l^s,B)} \right) \pm n \cdot \sigma(\sigma_{\kappa(C_l^s,B)}).$$

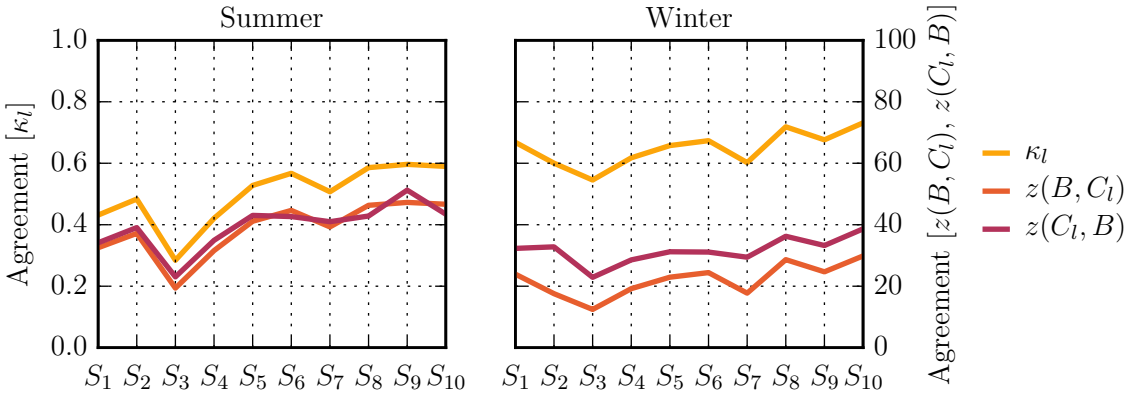
For precipitation data, these hold true in 43 out of 48 cases for $n = 2$, with the rest lying inside the $3\,\sigma$-environment. For temperature data, the equations are satisfied in all 24 cases for the $2\,\sigma$-environment.

The results of $z(B, C_l)$ and $z(C_l, B)$ generally closely match (see Figs. B.3 and B.4). In the rare instances where choice of surrogate method does make a difference, we found the reason to be variations of $\sigma_{\kappa(C_l^s,B)}$. The standard deviations are generally very small for our surrogates, so any changes in the denominator of Eq. (3.30) can have a large impact on the outcome. We believe these fluctuations to be of no physical importance, but they make the interpretation of $z(C_l, B)$ more difficult.

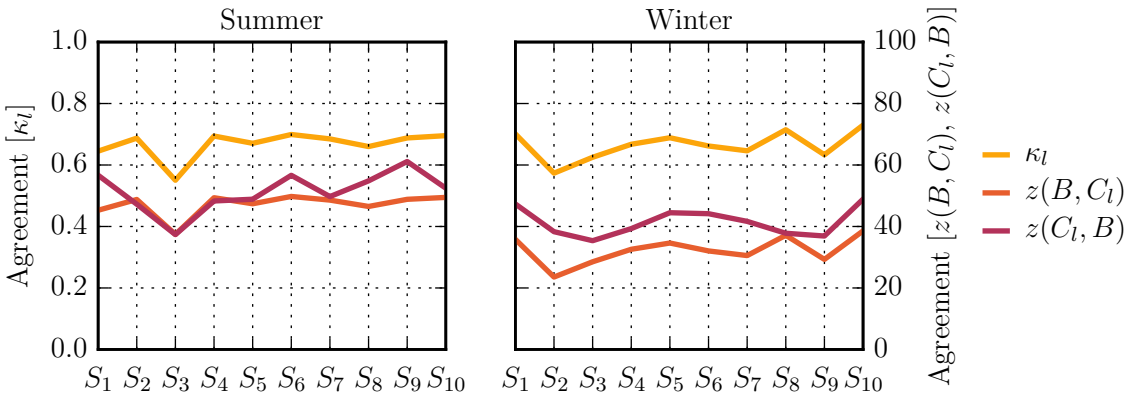


(a) Mean temperature, TCN50, Degree

Figure B.3.: See Fig. B.4 for description. These are the curves for mean temperature, TCN50 and node degree.



(a) Precipitation, ESN, Strength

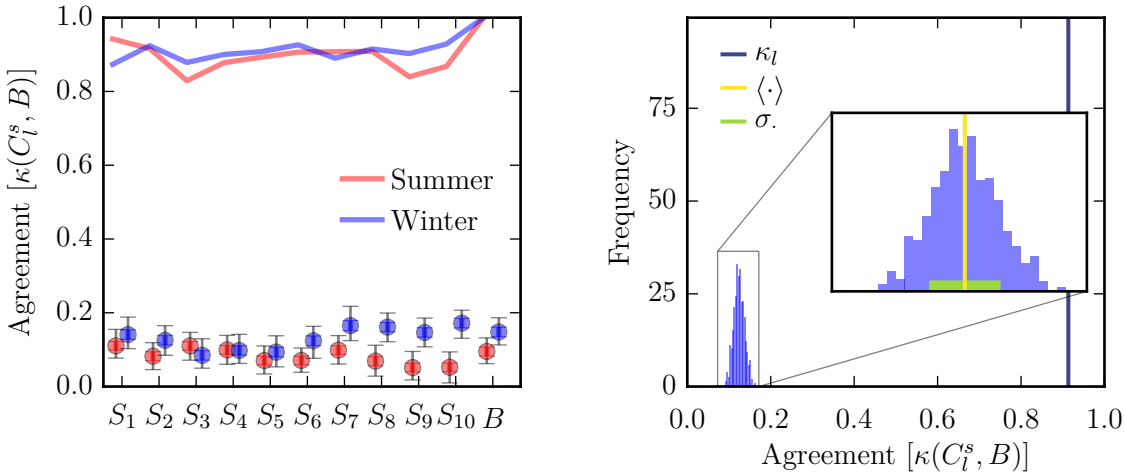


(b) Precipitation, ESN50, Degree

Figure B.4.: Comparison of z_l and $z(C_l, B)$ given precipitation data for degree (calculated from $\mathcal{A}$) and strength (derived from $\mathcal{W}$). Both z-types generally follow κ_l. In the last subfigure, an increased variability of $z(C_l, B)$ is visible.

Consequently, we consider B^s to be representative for all C_l^s, and can thus use Eq. (3.31) instead of Eq. (3.30) The main advantage here is the greatly reduced computational demand, since we only have to calculate 1000 surrogates instead of 11×1000. One disadvantage is that we lose some information, since the reference level that we test against remains the same for all chunks. We accept this trade-off, especially since our surrogates are only designed to test whether the whole system, especially the overall characteristics and structure of the data, influence the baseline κ. In any case, they are not sophisticated enough to highlight the finer differences between chunks.

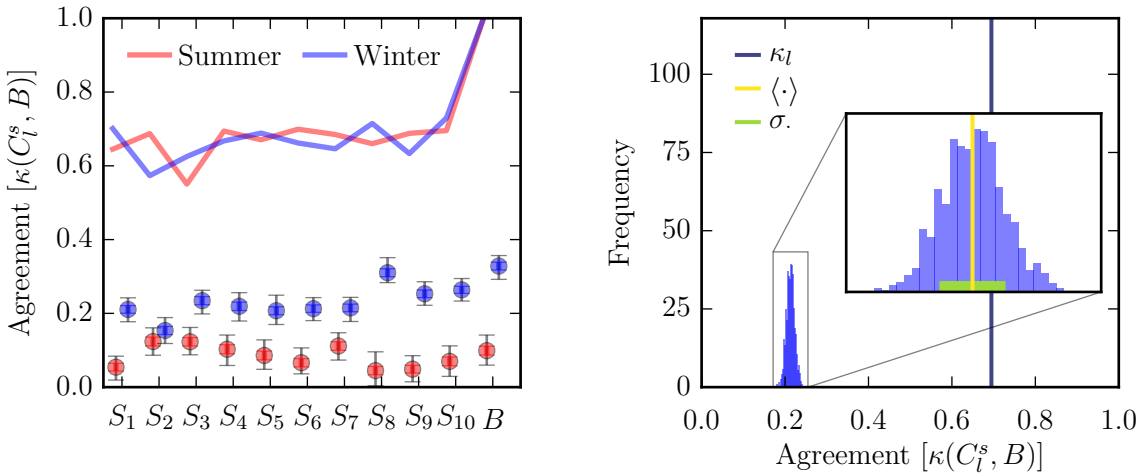
Surrogate distributions



(a) The distributions for all chunks and the basis data. Gray whiskers for the extremes, in color the mean and σ. The original κ-values as lines.

(b) The distribution for S_5^s. The surrogate κ follow a normal distribution with a relatively sharp peak. The original $\kappa(S_5, B)$ as a blue line.

Figure B.5.: Example of the distribution of $\kappa(C_l^s, B)$ for degree on a TCN50, calculated from the mean temperature winter data.



(a) The distributions for all chunks and the basis data. Gray whiskers for the extremes, in color the mean and σ. The original κ-values as lines.

(b) The distribution for S_5^s. The surrogate κ follow a normal distribution with a relatively sharp peak. The original $\kappa(S_5, B)$ is marked by a blue line.

Figure B.6.: Example of the distribution of $\kappa(C_l^s, B)$ for degree on a ESN50, calculated from the precipitation winter data.

Surrogate implementation

The following algorithm was used to generate the surrogate time series that we used to estimate the system's baseline κ.

Listing B.2: Pseudocode for the generation of surrogate time series from the original time series at some node.

```
for n in nodes:
    X = get_daily_timeseries(n)
    Y = empty_daily_timeseries()

    /*
        X = [x0, x1, x2, ...]
        X[[2, 1]] = [x2, x1]
        length of X, Y is approx. 50 * 365 days
    */

    months = [jan, feb, ...]

    for m in months:
        pool = []

        for x in X:
            if get_attribute(month, x) is m:
                append(pool, get_attribute(x, index))
            end
        end

        p_pool := get_random_permutation(pool)
        Y[pool] := X[p_pool]
    end
end
```

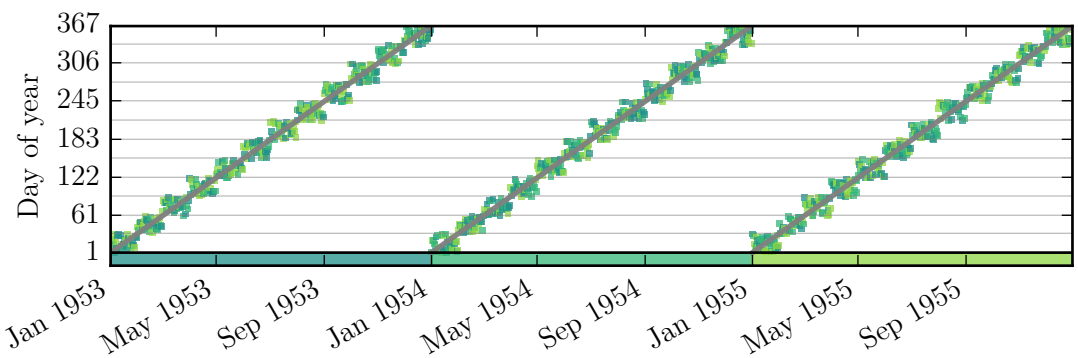
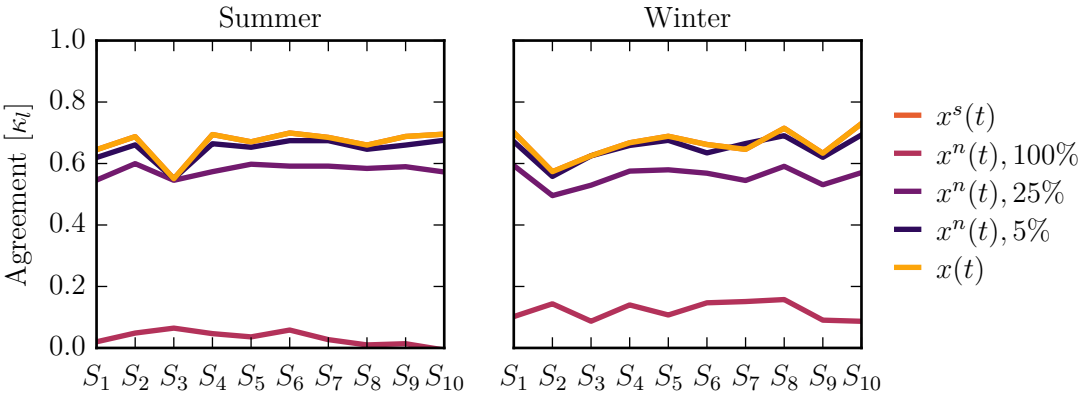
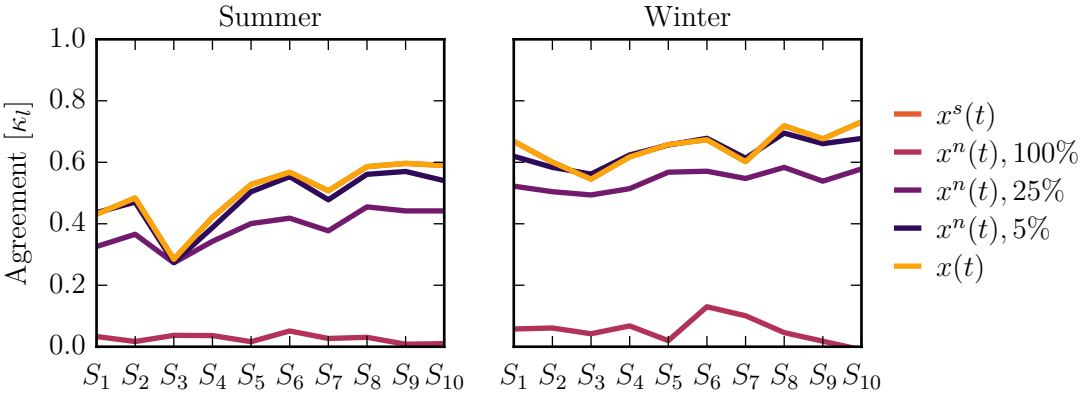


Figure B.7.: Illustration of the shuffle mechanism using months as pools and days as blocks. The gray line shows the source time series in order. The dots show the shuffled series: the y-coordinate is the original day of year, and the color indicates the original year. The gray lines mark months, illustrating the shuffle pools.

B.5. Noisy time series

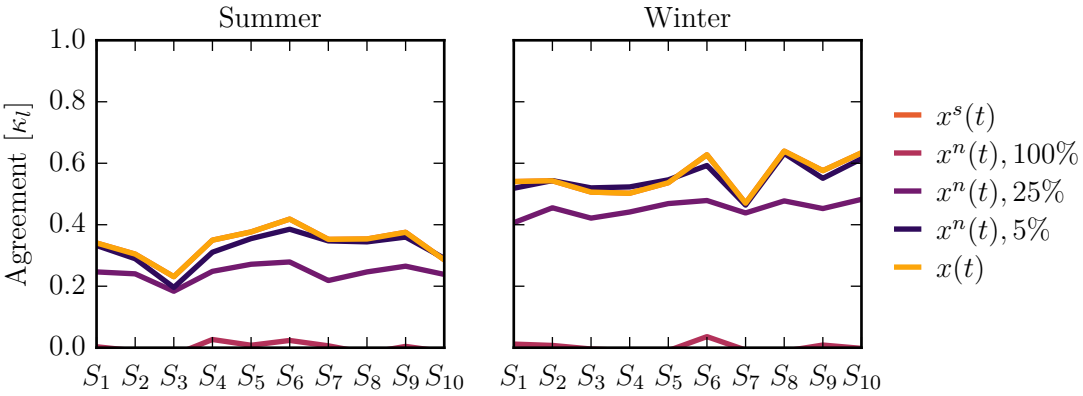


(a) ESN50, Degree.

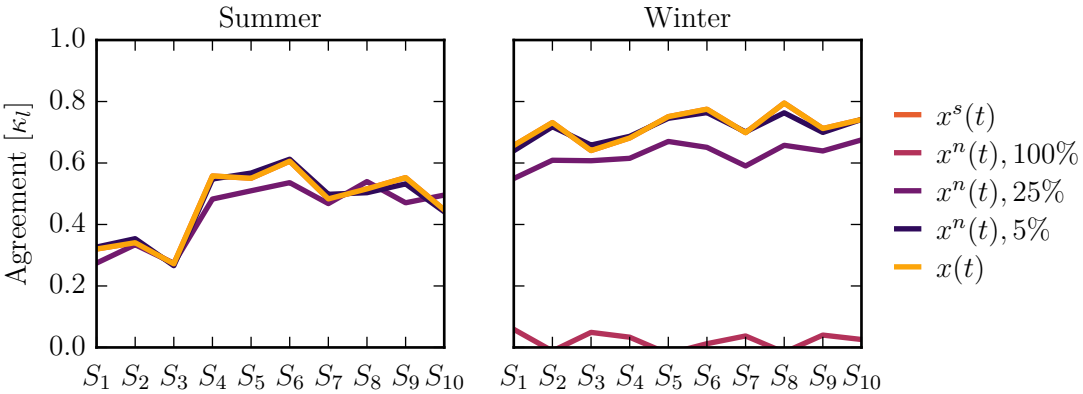


(b) ESN, Strength.

Figure B.8.: STARS performance compared to surrogates and noisy time series as measured with κ_l.



(a) ESN, Directionality ρ.



(b) ESN, Directionality φ.

Figure B.9.: STARS performance compared to surrogates and noisy time series as measured with κ_l.

C. Analysis

C.1. Comparison of STARS and basis data

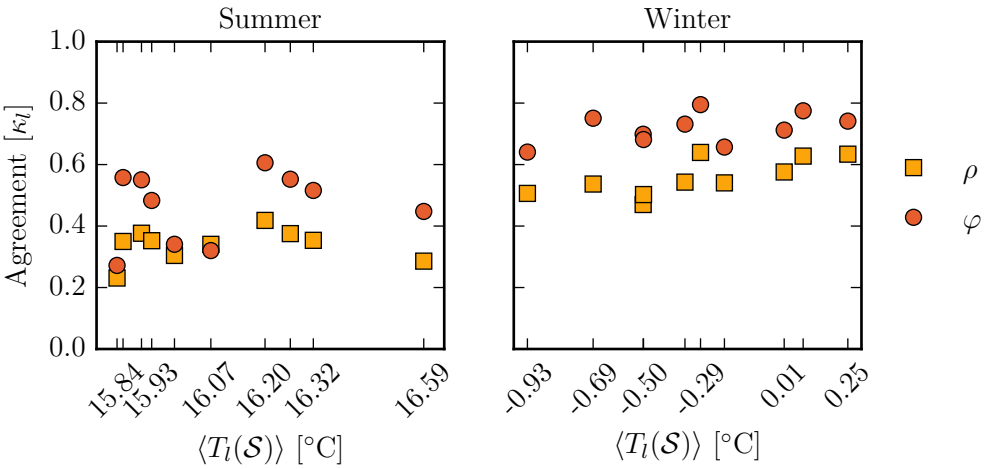


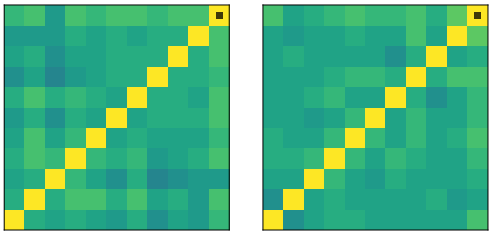
Figure C.1.: Directionality. Development of κ_l with chunk mean temperature.

C.2. Comparison matrices

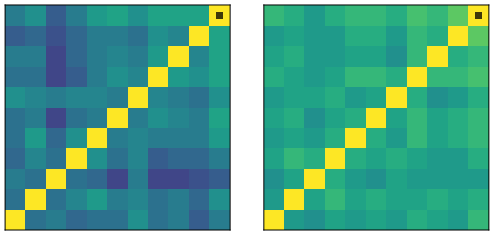
All measures by variable

The following figures each show the agreement for all network measures for a given climate variable, ordered either by time or by chunk seasonal mean temperature $\langle T_l(\mathcal{S})\rangle$. For degree, only the networks with $\varrho = 50\%$ are displayed, all other link densities can be found in the next subsection.

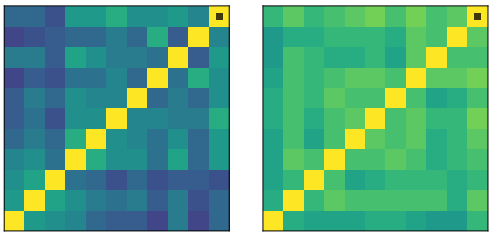
Precipitation



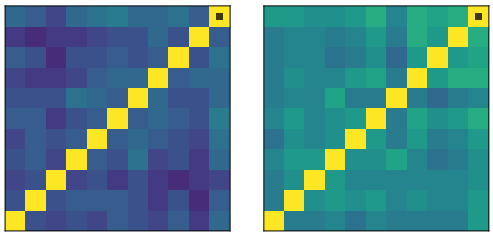
(a) P, ESN50, Degree.



(b) P, ESN, Strength.



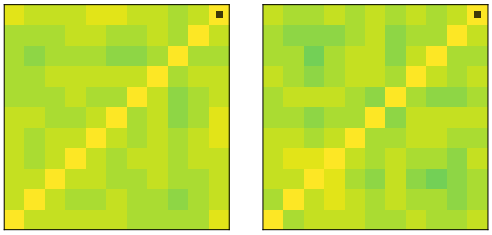
(c) P, ESN, Directionality φ.



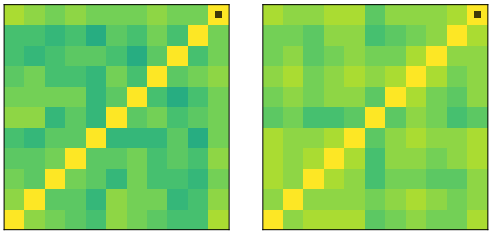
(d) P, ESN, Directionality ρ.

Figure C.2.: Precipitation, ordered by time. The row and column containing the basis data are marked by a dark rectangle. The left matrix in each subplot is summer, and the right matrix winter data.

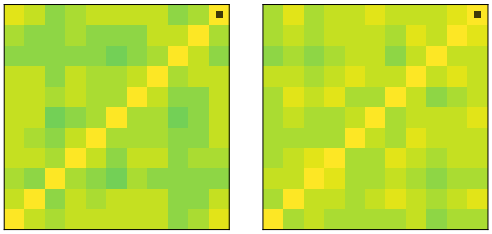
Temperature



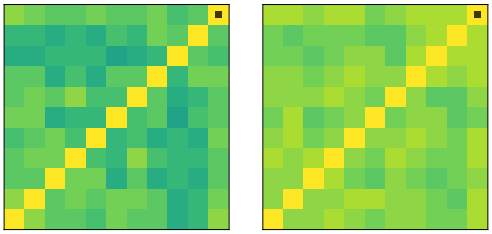
(a) $T_{\max}$, TCN50, Degree.



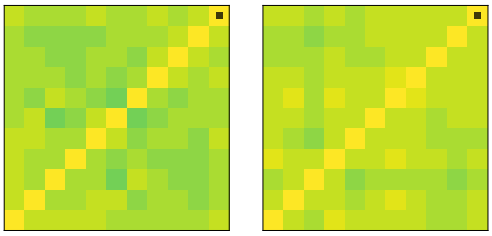
(b) $T_{\max}$, TCN, Strength.



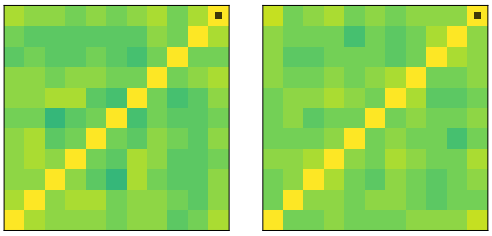
(c) T_{mean}, TCN50, Degree.



(d) T_{mean}, TCN, Strength.

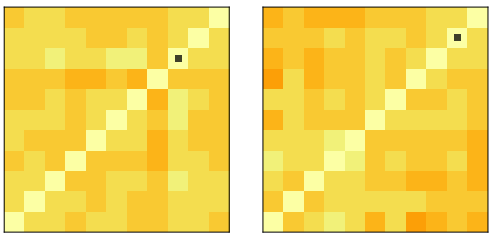


(e) $T_{\min}$, TCN50, Degree.

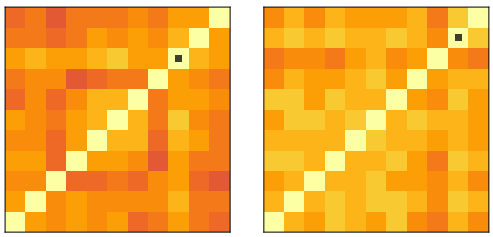


(f) $T_{\min}$, TCN, Strength.

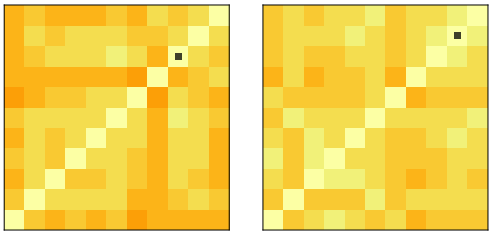
Figure C.3.: Temperature, ordered by time. The row and column containing the basis data are marked by a dark rectangle. The left matrix in each subplot is summer, and the right matrix winter data.



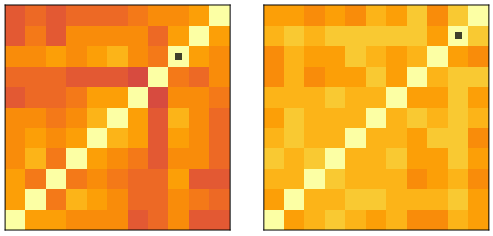
(a) $T_{\max}$, TCN50, Degree.



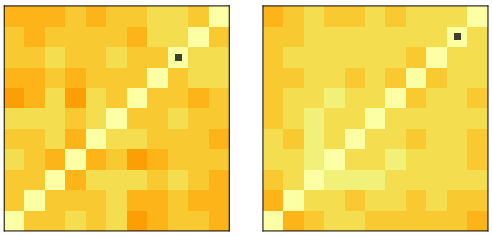
(b) $T_{\max}$, TCN, Strength.



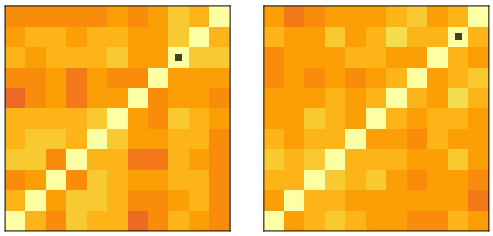
(c) T_{mean}, TCN50, Degree.



(d) T_{mean}, TCN, Strength.



(e) $T_{\min}$, TCN50, Degree.



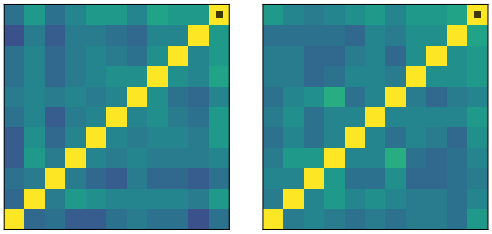
(f) $T_{\min}$, TCN, Strength.

Figure C.4.: Temperature, ordered by $\langle T_l(\mathcal{S})\rangle$. The row and column containing the basis data are marked by a dark rectangle. The left matrix in each subplot is summer, and the right matrix winter data.

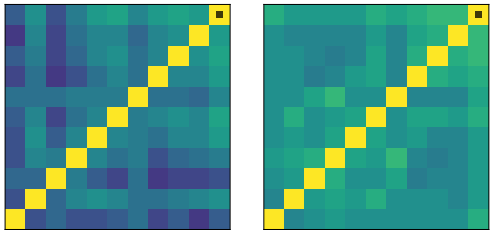
Degree by link density

The following figures each show the agreement for node degree for a given climate variable and for all link densities, ordered either by time or by chunk seasonal mean temperature $\langle T_l(\mathcal{S})\rangle$.

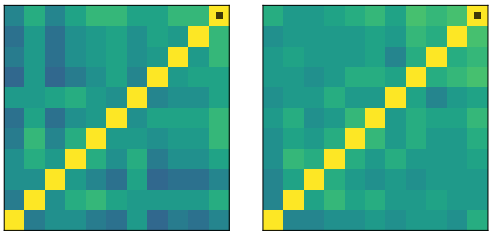
Precipitation



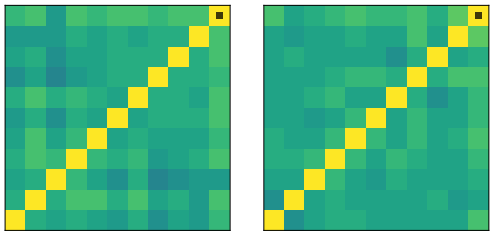
(a) P, ESN01, Degree.



(b) P, ESN10, Degree.

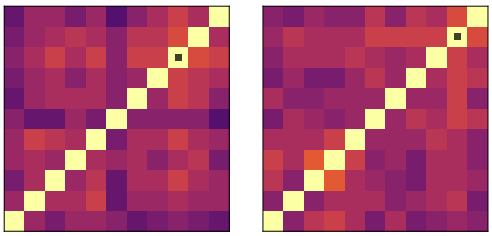


(c) P, ESN25, Degree.

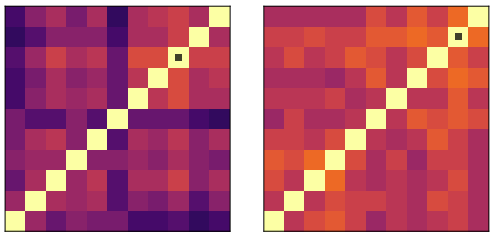


(d) P, ESN50, Degree.

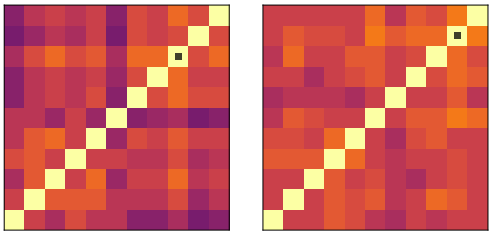
Figure C.5.: Precipitation, ordered by time. The row and column containing the basis data are marked by a dark rectangle. The left matrix in each subplot is summer, and the right matrix winter data.



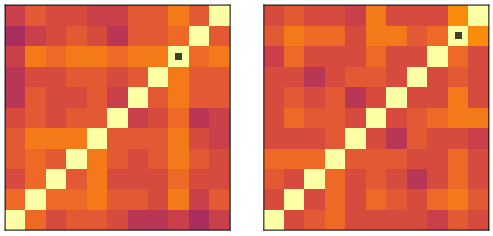
(a) P, ESN01, Degree.



(b) P, ESN10, Degree.



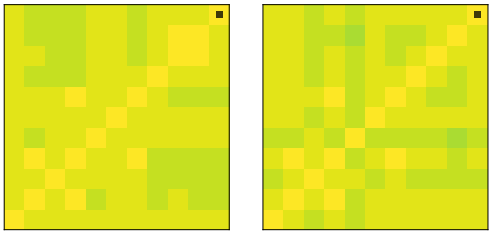
(c) P, ESN25, Degree.



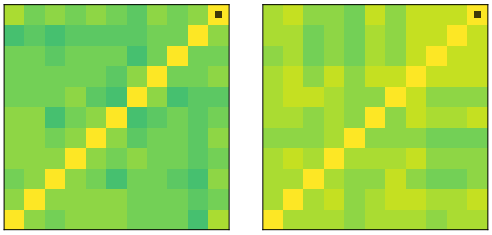
(d) P, ESN50, Degree.

Figure C.6.: Precipitation, ordered by $\langle T_l(\mathcal{S})\rangle$. The row and column containing the basis data are marked by a dark rectangle. The left matrix in each subplot is summer, and the right matrix winter data.

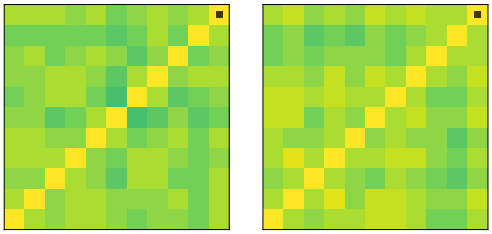
Minimum temperature



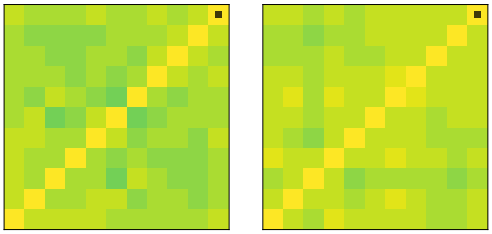
(a) $T_{\min}$, TCN01, Degree.



(b) $T_{\min}$, TCN10, Degree.

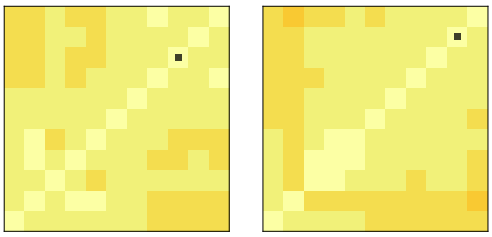


(c) $T_{\min}$, TCN25, Degree.

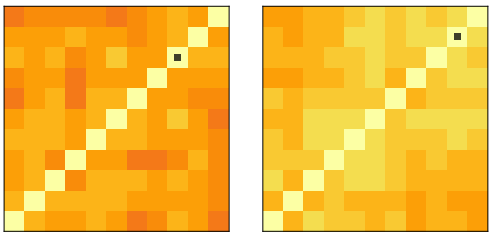


(d) $T_{\min}$, TCN50, Degree.

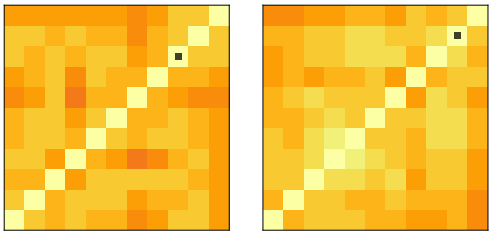
Figure C.7.: Minimum temperature, ordered by time. The row and column containing the basis data are marked by a dark rectangle. The left matrix in each subplot is summer, and the right matrix winter data.



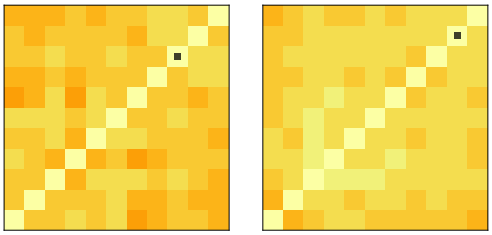
(a) $T_{\min}$, TCN01, Degree.



(b) $T_{\min}$, TCN10, Degree.



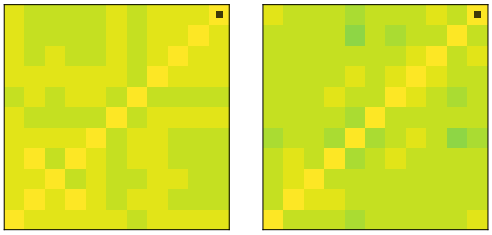
(c) $T_{\min}$, TCN25, Degree.



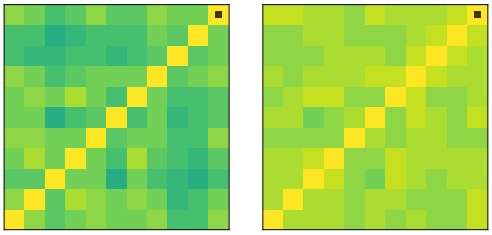
(d) $T_{\min}$, TCN50, Degree.

Figure C.8.: Minimum temperature, ordered by $\langle T_l(\mathcal{S})\rangle$. The row and column containing the basis data are marked by a dark rectangle. The left matrix in each subplot is summer, and the right matrix winter data.

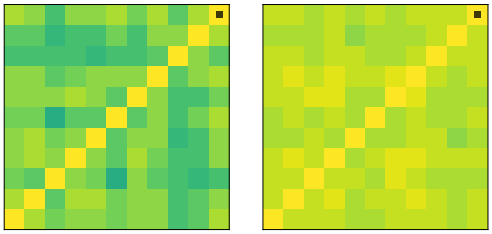
Mean temperature



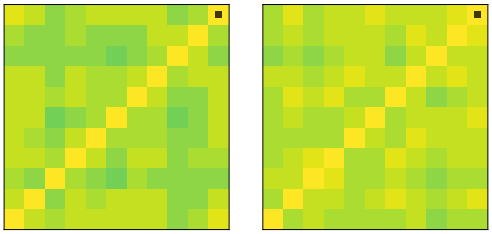
(a) T_{mean}, TCN01, Degree.



(b) T_{mean}, TCN10, Degree.

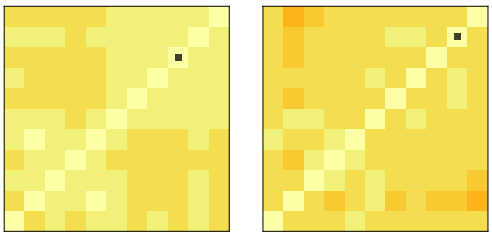


(c) T_{mean}, TCN25, Degree.

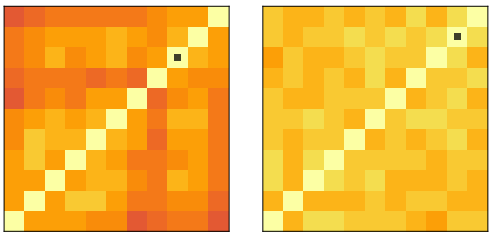


(d) T_{mean}, TCN50, Degree.

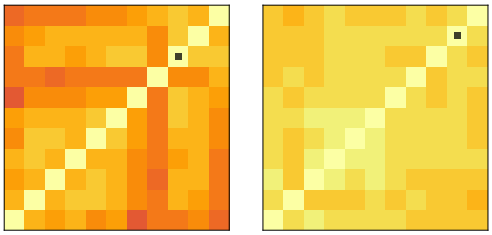
Figure C.9.: Mean temperature, ordered by time. The row and column containing the basis data are marked by a dark rectangle. The left matrix in each subplot is summer, and the right matrix winter data.



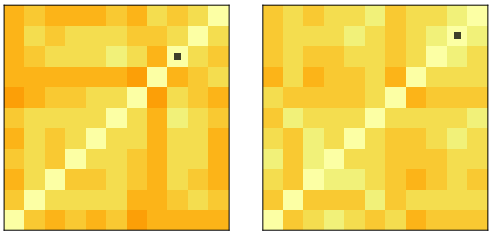
(a) T_{mean}, TCN01, Degree.



(b) T_{mean}, TCN10, Degree.



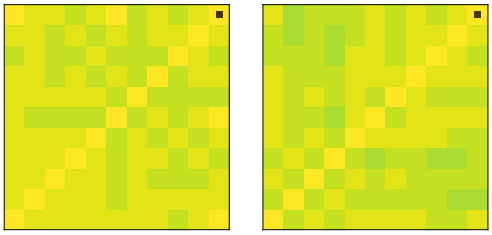
(c) T_{mean}, TCN25, Degree.



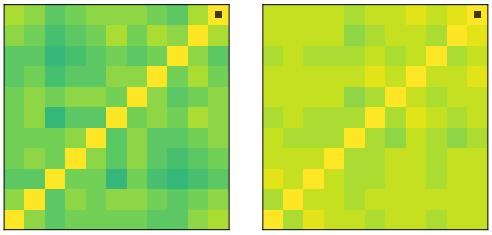
(d) T_{mean}, TCN50, Degree.

Figure C.10.: Mean temperature, ordered by $\langle T_l(\mathcal{S})\rangle$. The row and column containing the basis data are marked by a dark rectangle. The left matrix in each subplot is summer, and the right matrix winter data.

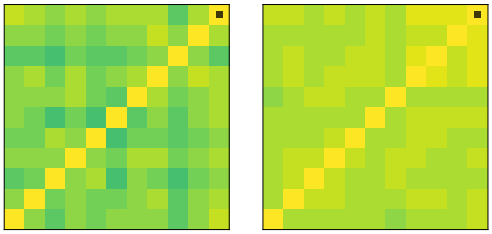
Maximum temperature



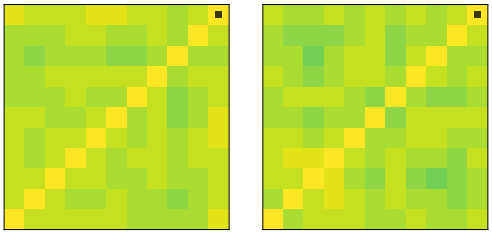
(a) $T_{\max}$, TCN01, Degree.



(b) $T_{\max}$, TCN10, Degree.

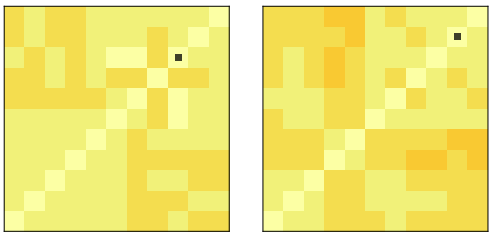


(c) $T_{\max}$, TCN25, Degree.

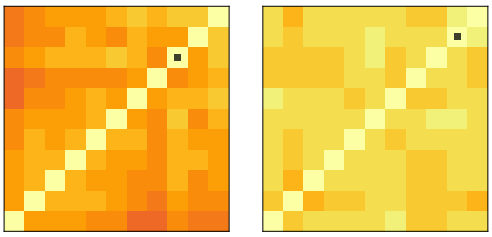


(d) $T_{\max}$, TCN50, Degree.

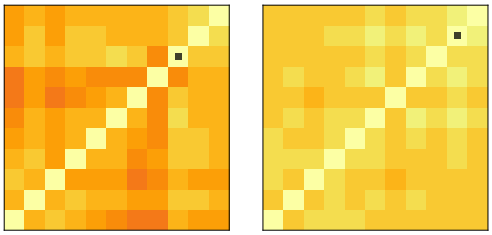
Figure C.11.: Maximum temperature, ordered by time. The row and column containing the basis data are marked by a dark rectangle. The left matrix in each subplot is summer, and the right matrix winter data.



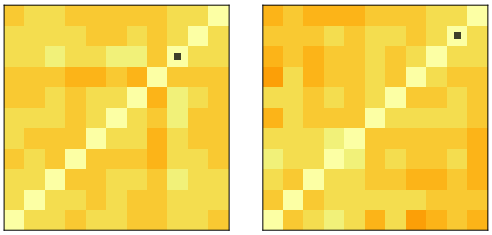
(a) $T_{\max}$, TCN01, Degree.



(b) $T_{\max}$, TCN10, Degree.



(c) $T_{\max}$, TCN25, Degree.



(d) $T_{\max}$, TCN50, Degree.

Figure C.12.: Maximum temperature, ordered by $\langle T_l(\mathcal{S})\rangle$. The row and column containing the basis data are marked by a dark rectangle. The left matrix in each subplot is summer, and the right matrix winter data.

C.3. Network measure distributions

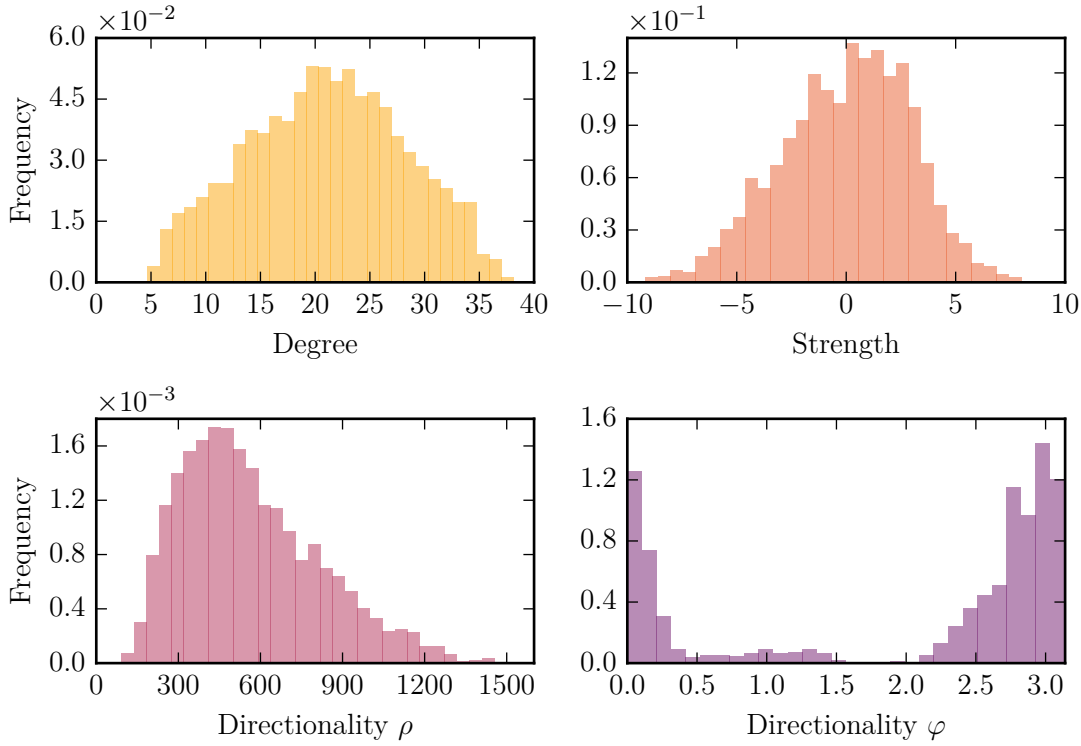


Figure C.13.: Typical distributions of network measure scores for an ESN. Displayed is the data of S_5.

C.4. Linear regressions

Table C.1.: Mean and standard deviation of κ_l per network measure. The last columns give the coefficients of a linear regression of $\kappa_l = a \cdot \langle T_l(\mathcal{S}) \rangle + b$ and the respective coefficient of determination r^2.

Measure	Season	Variable	Link density	$\langle \kappa \rangle$	σ_κ	a	b	r^2
k	Summer	T_{max}	50	0.91	0.02	-0.027	1.352	0.09
		T_{mean}		0.89	0.03	-0.005	0.974	0.00
		T_{min}		0.88	0.02	0.059	-0.077	0.38
		P		0.67	0.04	0.069	-0.445	0.14
	Winter	T_{max}		0.89	0.02	0.020	0.893	0.12
		T_{mean}		0.90	0.02	0.031	0.914	0.32
		T_{min}		0.89	0.01	0.021	0.901	0.29
		P		0.66	0.04	0.044	0.678	0.11
s	Summer	T_{max}	-	0.79	0.03	-0.014	1.025	0.02
		T_{mean}		0.76	0.02	-0.033	1.287	0.09
		T_{min}		0.82	0.03	0.046	0.085	0.12
		P		0.50	0.09	0.294	-4.230	0.53
	Winter	T_{max}		0.83	0.03	-0.019	0.828	0.04
		T_{mean}		0.84	0.02	0.007	0.843	0.01
		T_{min}		0.82	0.03	0.029	0.832	0.10
		P		0.65	0.05	0.123	0.688	0.60
φ	Summer	P		0.46	0.11	0.107	-1.260	0.05
	Winter	P		0.72	0.05	0.063	0.738	0.21
ρ	Summer	P		0.34	0.05	0.020	0.014	0.01
	Winter	P		0.56	0.06	0.121	0.596	0.54

Table C.2.: Mean and standard deviation of κ_l for k per link density. The last columns give the coefficients of a linear regression of $\kappa_l = a \cdot \langle T_l(\mathcal{S}) \rangle + b$ and the respective coefficient of determination r^2.

Measure	Season	Variable	Link density	$\langle\kappa\rangle$	σ_κ	a	b	r^2
k	Summer	T_{max}	1	0.93	0.02	0.032	0.421	0.19
			10	0.81	0.04	0.062	-0.186	0.12
			25	0.83	0.04	0.015	0.586	0.01
			50	0.91	0.02	-0.027	1.352	0.09
		T_{mean}	1	0.92	0.02	0.069	-0.197	0.77
			10	0.78	0.04	0.073	-0.403	0.20
			25	0.82	0.05	0.105	-0.879	0.25
			50	0.89	0.03	-0.005	0.974	0.00
		T_{min}	1	0.93	0.02	0.053	0.078	0.42
			10	0.80	0.02	0.009	0.655	0.01
			25	0.84	0.02	0.016	0.592	0.02
			50	0.88	0.02	0.059	-0.077	0.38
		P	1	0.50	0.07	0.158	-2.047	0.29
			10	0.48	0.10	0.242	-3.413	0.30
			25	0.59	0.07	0.122	-1.377	0.18
			50	0.67	0.04	0.069	-0.445	0.14
	Winter	T_{max}	1	0.91	0.02	0.036	0.923	0.32
			10	0.90	0.02	0.018	0.908	0.12
			25	0.90	0.02	0.045	0.914	0.54
			50	0.89	0.02	0.020	0.893	0.12
		T_{mean}	1	0.91	0.02	0.031	0.918	0.29
			10	0.86	0.03	0.054	0.881	0.51
			25	0.89	0.02	0.030	0.901	0.41
			50	0.90	0.02	0.031	0.914	0.32
		T_{min}	1	0.93	0.02	0.045	0.946	0.69
			10	0.86	0.04	0.099	0.890	0.73
			25	0.87	0.03	0.058	0.887	0.53
			50	0.89	0.01	0.021	0.901	0.29
		P	1	0.50	0.05	0.111	0.533	0.70
			10	0.58	0.06	0.140	0.628	0.76
			25	0.62	0.05	0.118	0.655	0.58
			50	0.66	0.04	0.044	0.678	0.11

Table C.3.: Mean and standard deviation of κ_l for k per link density. The last columns give the coefficients of a linear regression of $\kappa_l = a \cdot \langle l \rangle + b$ and the respective coefficient of determination r^2.

Measure	Season	Variable	Link density	$\langle\kappa\rangle$	σ_κ	a	b	r^2
k	Summer	T_{max}	1	0.93	0.02	-0.001	0.936	0.03
			10	0.81	0.04	-0.001	0.810	0.01
			25	0.83	0.04	-0.004	0.854	0.11
			50	0.91	0.02	-0.005	0.931	0.46
		T_{mean}	1	0.92	0.02	0.003	0.908	0.21
			10	0.78	0.04	0.001	0.772	0.00
			25	0.82	0.05	0.003	0.804	0.03
			50	0.89	0.03	-0.004	0.908	0.13
		T_{min}	1	0.93	0.02	0.002	0.922	0.13
			10	0.80	0.02	-0.003	0.812	0.11
			25	0.84	0.02	-0.001	0.845	0.01
			50	0.88	0.02	0.002	0.870	0.07
		P	1	0.50	0.07	0.016	0.429	0.43
			10	0.48	0.10	0.024	0.366	0.47
			25	0.59	0.07	0.015	0.523	0.40
			50	0.67	0.04	0.006	0.640	0.17
	Winter	T_{max}	1	0.91	0.02	0.004	0.896	0.22
			10	0.90	0.02	0.001	0.898	0.01
			25	0.90	0.02	0.004	0.882	0.28
			50	0.89	0.02	-0.001	0.893	0.04
		T_{mean}	1	0.91	0.02	-0.001	0.911	0.01
			10	0.86	0.03	-0.001	0.868	0.01
			25	0.89	0.02	0.001	0.889	0.01
			50	0.90	0.02	0.003	0.891	0.24
		T_{min}	1	0.93	0.02	0.002	0.923	0.08
			10	0.86	0.04	0.006	0.834	0.16
			25	0.87	0.03	0.002	0.862	0.03
			50	0.89	0.01	0.001	0.891	0.04
		P	1	0.50	0.05	0.010	0.453	0.39
			10	0.58	0.06	0.012	0.530	0.39
			25	0.62	0.05	0.014	0.554	0.60
			50	0.66	0.04	0.006	0.636	0.16

D. Other

D.1. Acknowledgments

I would like to express my gratitude to the people at PIK, especially Norbert Marwan and Aljoscha Rheinwalt, for their help with this thesis, and to Jan Volkholz for providing the dataset I analyzed.

In addition, I am grateful to Aljoscha Rheinwalt for two software libraries that compute event synchronization scores at zero lag, and the spatially corrected strength and directionality measures. Examining the code of the `pyunicorn` project (see [42]) helped me immensely in understanding climate networks, and I would like to thank its creators as well. More generally, I am indebted to the people behind the Python language, especially the open-source packages `NumpPy`, `SciPy`, `Matplotlib`, and `Pandas`, and to those behind the whole LaTeX-ecosystem.

Lastly, I want to thank my family, and in particular my parents, for their continuous support, fruitful discussions and encouraging words over the last months.

D.2. Statement of authorship

I declare that I completed this thesis on my own and that information which has been directly or indirectly taken from other sources has been noted as such.[1]

Neither this nor a similar work has been presented to an examination committee.

. . . . . .

Malte Viefhues

[1] The text of this declaration has been copied from `http://www2.informatik.hu-berlin.de/sv/lehre/hinweise.shtml`.