Supplement to 'A Neural Network-Based Adaptive Cut-Off Approach to Normality Testing for Dependent Data'

Minwoo Kim
Department of Statistics, Pusan National University
Marc G. Genton and Raphaël Huser
Statistics program, King Abdullah University of Science and Technology
Stefano Castruccio*
Department of Applied and Computational Mathematics and Statistics, University of Notre Dame

A Section 3: Simulation study

Simulation results with $\nu = 1$

In this section, we present results of numerical experiments with $\nu = 1.0$. Data are simulated on a two dimensional unit square regular grid of size 60×60 from a baseline zero mean, isotropic random field with Matérn function. We choose $n_{\beta;\text{train}} = 30$ equally spaced values of β between 0 and $\beta_{\max} = 0.175$ (including both endpoints) in the training set, spanning from zero to strong dependence related to the effective range of 0.7 on a unit square. In the test set we choose $n_{\beta;\text{test}} = 50$ equally spaced values of β from 0 to $\beta_{\max}$. The sets of β in the training set and testing set are denoted by $\mathcal{B}_{\text{train}}$ and $\mathcal{B}_{\text{test}}$, respectively, such that $|\mathcal{B}_{\text{train}}| = n_{\beta;\text{train}}$ and $|\mathcal{B}_{\text{test}}| = n_{\beta;\text{test}}$. Non-normal distributions in the training and testing set were created by applying a signed power transformation to the baseline Matérn Gaussian random field. Specifically, for an exponent parameter p, a value z was transformed to $f(z;p) = z^p \text{sign}(z)$, for values of p in the set $\mathcal{P}_{\text{train}} = \{1.2, 1.4, 1.6, 1.8\}$ in the training set, and in the set $\mathcal{P}_{\text{test}} = \{1.1, 1.2, \ldots, 2.0\}$ in the testing set. We denote by $|\mathcal{P}_{\text{train}}| = n_{p;\text{train}}$ and $|\mathcal{P}_{\text{test}}| = n_{p;\text{test}}$, and we generate $n_{\text{sample}} = 200$ sample points for each combination of (β, p) in the case of non-normal data. Therefore, the training set contains $n_{\beta;\text{train}} \times n_{p;\text{train}} \times n_{\text{sample}} = 24,000$ (non-normal) data points, while the testing set contains $n_{\beta;\text{test}} \times n_{p;\text{test}} \times n_{\text{sample}} = 100,000$ (non-normal) data points. For the null hypothesis, i.e. normal data with $p = 1$, we generate an equivalent number

*Corresponding author, scastruc@nd.edu

of samples, i.e, the training set contains $24,000$ points, while the testing set contains $100,000$ points.

We use $m = 6$ inputs: the four test statistics of the normality tests in Section 2.1 of the main manuscript along with the sample skewness and kurtosis. We rely on a neural networks with $L = 2$ layers and with $n_1 = 256$ and $n_2 = 128$ nodes. In each of the L layers, 30% of nodes are randomly removed during each training step and inference is performed by minimizing the binary cross-entropy logarithmic loss (1) in the main manuscript. Cut-off functions for neural network and linear classifiers from non-parametric kernel regression are shown in Figure S.1.

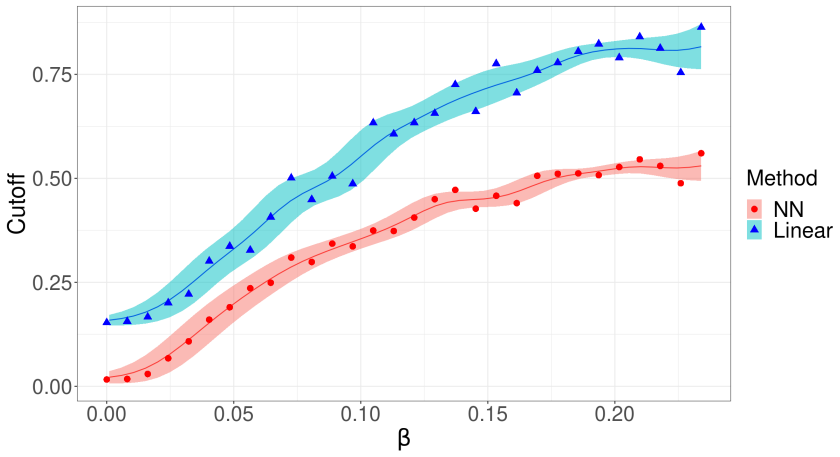
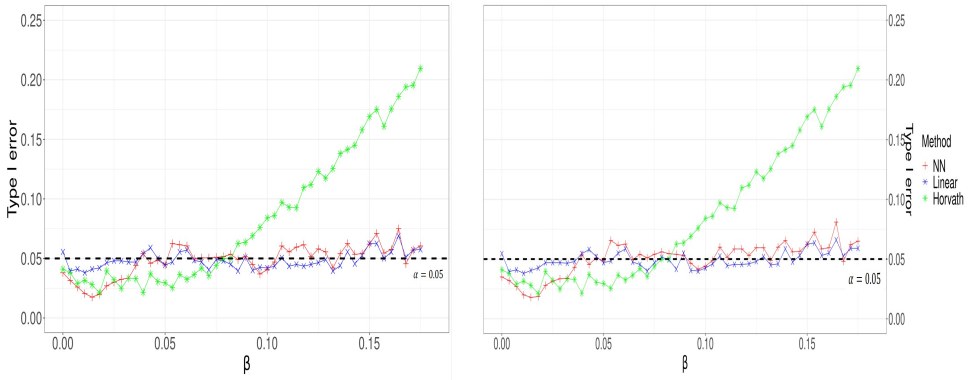


Figure S.1: Simulation study: Non-parametric Gaussian kernel regressions as defined in (4) of the main manuscript with $h = 0.3$ for neural network (red) and linear (blue) classifiers. On the x-axis are the range parameter β of the Matérn covariance, while on the y-axis the predicted cut-off and corresponding pointwise 95% confidence interval are represented by solid lines and bands respectively. The other two parameters are fixed at $\sigma^2 = 1$ and $\nu = 1$.

Type I error comparison

We compare the Type I errors of the three tests - neural network, linear classifier, and Horváth et al. (2020)'s method - for both scenarios where the values of $\beta \in \mathcal{B}_{test}$ are known (Figure S.2-(a)) and where their values are unknown(Figure S.2-(b)). The unknown values of β are estimated using the software `ExaGeoStatR` (Abdulah et al., 2023)



(a) (b)

Figure S.2: Panel (a): Type I errors for neural network (red), linear classifier (blue), and Horváth et al. (2020)'s method (green) assuming that the parameters $\beta \in \mathcal{B}_{\text{test}}$ are known where $|\mathcal{B}_{\text{test}}| = 50$ and the other two parameters of a Matérn covariance function are fixed, $\sigma^2 = 1$ and $\nu = 1.0$; Panel (b): Same as (a) when the parameters β on the x-axis and σ^2 are estimated by maximum likelihood estimates and the smoothness parameter is fixed, $\nu = 1$. The black dashed line in each panel represents the 5% Type I error.

Power comparison

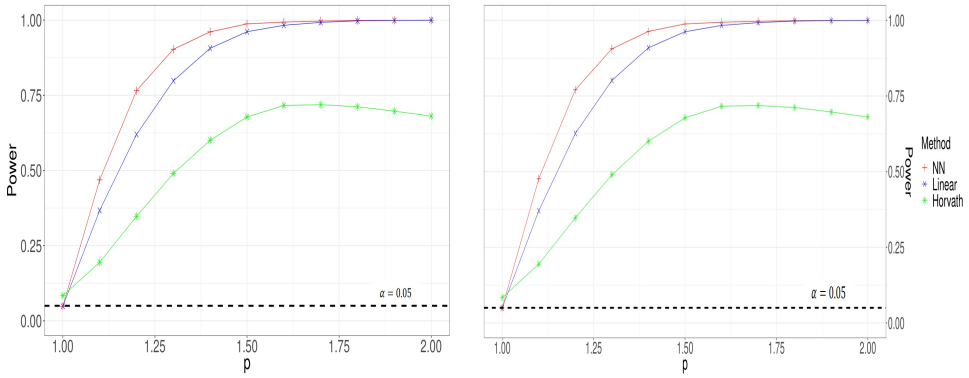
In this subsection, overall powers for our adaptive cut-off approaches and the Horváth et al. (2020)'s method, are described with $\nu = 1$. Figure S.3 shows the power curves as a function of the departure from normality, measured by the exponent p. Each curve is computed as an average across all choices of dependence parameters $\beta \in \mathcal{B}_{\text{test}}$ assuming that they are known (See Panel (a)) or estimated (See Panel (b)).

Misspecified smoothness parameter

In this subsection, we provide cases of misspecified parameter ν. Specifically, we train the neural network and linear models using the data generated with $\nu = 1$, while the actual test data is generated with $\nu = 0.5$, and vice versa. In Figure S.4, Type I errors and powers for both misspecification scenarios are presented.

Varying smoothness parameters with $\beta \in \{0.05, 0.2\}$

When we use a lower range parameter $\beta = 0.05$ or a higher range parameter $\beta = 0.2$, the maximum values of ν corresponding to the effective range of 0.7 on a unit square are $\nu_{max} = 15.9$ or $\nu_{max} = 0.73$, respectively. For each option of $\beta \in \{0.05, 0.2\}$, we choose $n_{\nu;\text{train}} = 30$ equally spaced values of ν including the minimum 0.1 and the maximum $\nu_{\text{max}} = 0.73$ in $\mathcal{V}_{\text{train}} =$



(a) (b)

Figure S.3: Panel (a): Averaged powers across all choices of $\beta \in \mathcal{B}_{\text{test}}$ for neural network (red), linear classifier (blue), and Horváth et al. (2020)'s method (green) given a value of p on the x-axis assuming that the parameters $\beta \in \mathcal{B}_{\text{test}}$ are known where $|\mathcal{B}_{\text{test}}| = 50$ and the other two parameters of a Matérn covariance function are fixed, $\sigma^2 = 1$ and $\nu = 1$; Panel (b): Same as (a) when the parameters (β, σ^2) are estimated by maximum likelihood estimates and the smoothness parameter is fixed, $\nu = 1$. The black dashed line in each panel represents the power of 5%.

$\{\nu_1, \ldots, \nu_{n_{\nu;\text{train}}}\}$, while $n_{\nu;\text{test}} = 50$ equally spaced values are chosen in $\mathcal{V}_{\text{test}}$ including the same endpoints, 0.1 and $\nu_{\max}$. Other settings are the same as those in Section 3.4 of the main manuscript. The resulting Type I errors and powers for the neural network, the linear classifier and the method in Horváth et al. (2020) are shown in Figure S.5.

Other non-normal distributions and covariance

Let the M-dimensional vector $\boldsymbol{y}$ and u be independent and distributed as a zero-mean multivariate normal with a covariance matrix $\boldsymbol{\Sigma}$ and chi-squared with a degree of freedom r, respectively. Then, the random vector $\boldsymbol{x} = \boldsymbol{y}/\sqrt{u/r}$ follows a zero-mean multivariate t-distribution whose density function is:

$$\frac{\Gamma\left[(r+M)/2\right]}{\Gamma[r/2] r^{M/2} \pi^{M/2} |\boldsymbol{\Sigma}|^{1/2}} \left(1 + \frac{1}{r} \boldsymbol{x}^\top \boldsymbol{\Sigma} \boldsymbol{x}\right)^{-(r+M)/2},$$

where $\boldsymbol{\Sigma}$ is a scale matrix and covariance matrix of $\boldsymbol{x}$ is $\text{Cov}(\boldsymbol{x}) = \frac{r}{r-2}\boldsymbol{\Sigma}$ if $r > 2$.

We use multivariate t-distributions as another choice of non-normal distributions. In the training set, multivariate t-distributed data are generated with $r \in \{1, 3, 5, 7\}$ while the data in the testing set are generated with $r \in \{1, 2, 3, 4, 5, 6, 7, 8\}$. The choices of r values in the training and testing

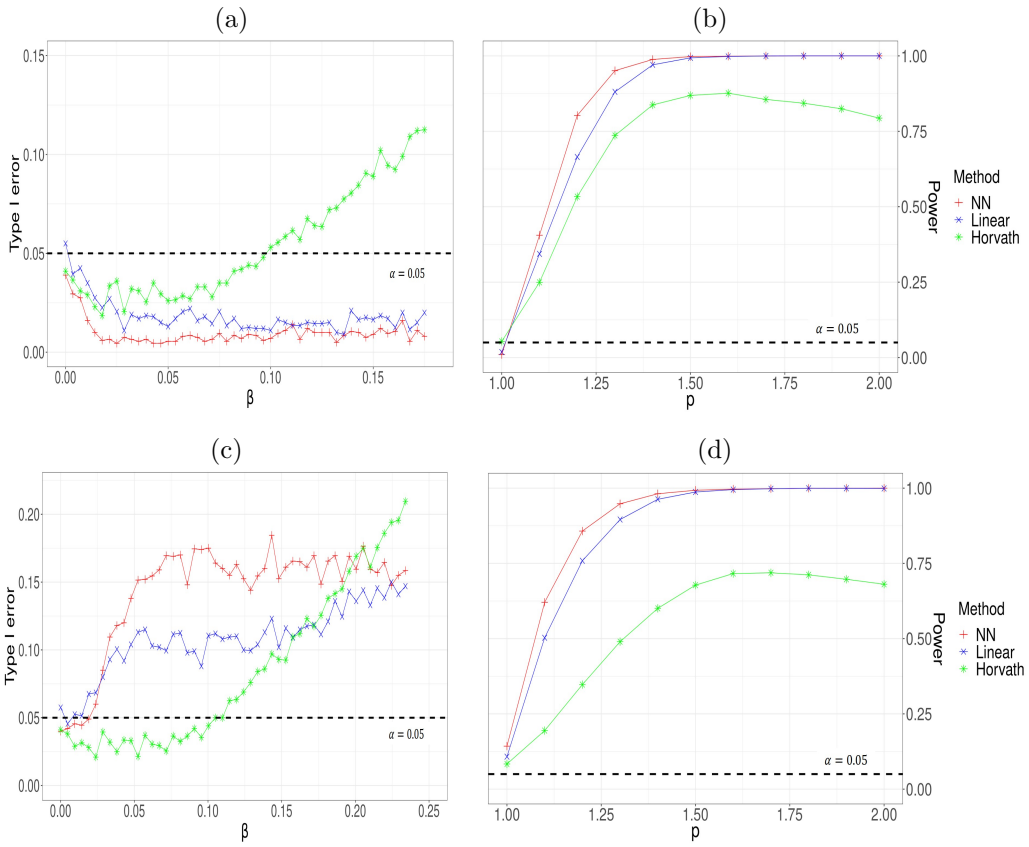


Figure S.4: In all the panels, we depict each method using distinct colors - neural network (red), linear classifier (blue), and the method proposed by Horváth et al. (2020) (green). Panel (a): Type I errors of the three testing methods when the neural network and linear classifiers are trained using $\nu = 1$, but the test data is generated using $\nu = 0.5$; Panel (b): Powers of the three testing methods when the neural network and linear classifiers are trained using $\nu = 1$, but the test data is generated using $\nu = 0.5$; Panel (c) and (d): Same as (a) and (b) respectively when the neural network and linear classifiers are trained using $\nu = 0.5$, but the test data is generated using $\nu = 1$.

set enable the neural networks to develop interpolation and extrapolation. To generate normal data, as in Section 3 of the main paper, we consider Matérn covariance functions with $\beta \in \mathcal{B}_{\text{train}}$ for training set and $\beta \in \mathcal{B}_{\text{test}}$ for training set when the other two parameters are fixed, $\sigma^2 = 1$, $\nu = 0.5$. Noting that $r = \infty$ indicates normality, the hypotheses H_0 and H_1 are

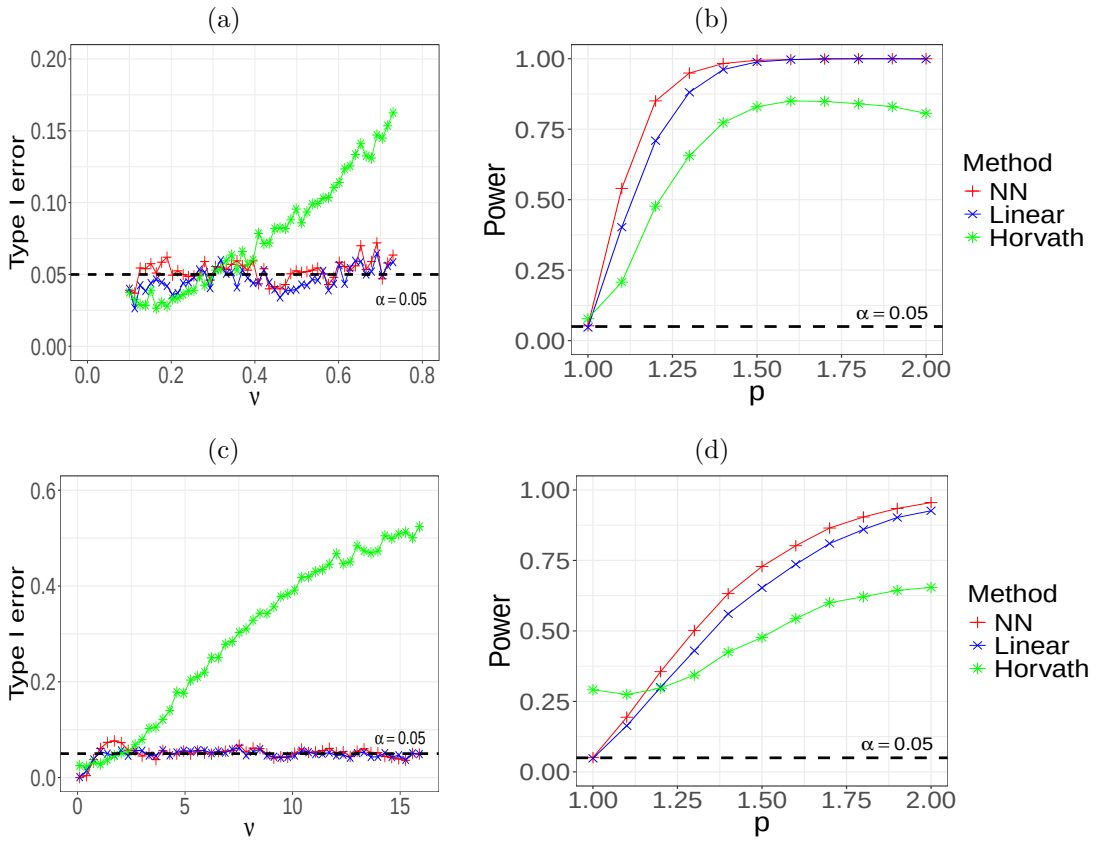


Figure S.5: Panel (a) and (c): Type I errors for neural network (red), linear classifier (blue), and Horváth et al. (2020)'s method (green) when the variance parameter of a Matérn covariance function is fixed, $\sigma^2 = 1$. The panels (a) and (c) correspond to $\beta = 0.2$ and $\beta = 0.05$, respectively. Panel (b) and (d): Averaged powers across all choices of $\nu \in \mathcal{V}_{\text{test}}$ for neural network (red), linear classifier (blue), and Horváth et al. (2020)'s method (green) given a value of exponent p (i.e., non-normality parameter) on the x-axis when the variance parameter of a Matérn covariance function is fixed, $\sigma^2 = 1$. The panels (b) and (d) correspond to $\beta = 0.2$ and $\beta = 0.05$, respectively.

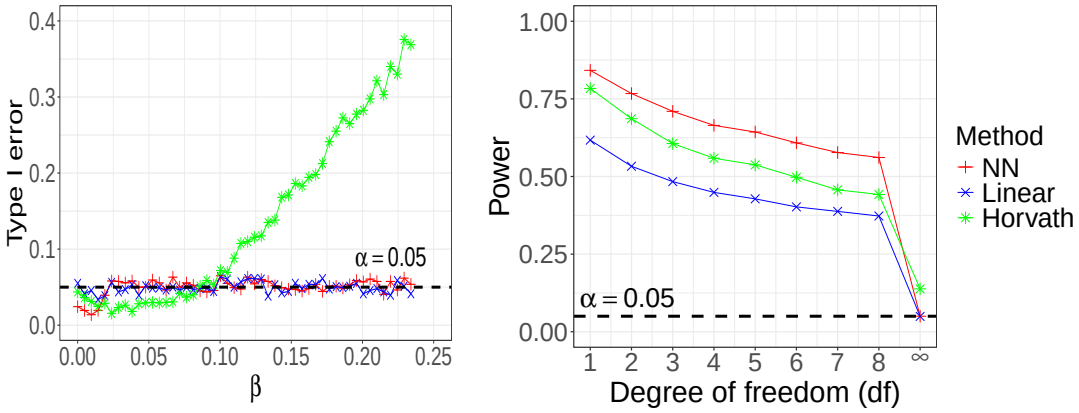
defined as follows:

$$H_0 : r = \infty \text{ and } \beta \in \mathcal{B}_{\text{train}},$$
$$H_1 : r \in \{1, 3, 5, 7\} \text{ and } \beta \in \mathcal{B}_{\text{train}},$$

for training set and

$$H_0 : r = \infty \text{ and } \beta \in \mathcal{B}_{\text{test}},$$
$$H_1 : r \in \{1, 2, 3, 4, 5, 6, 7, 8\} \text{ and } \beta \in \mathcal{B}_{\text{test}},$$

for testing set.



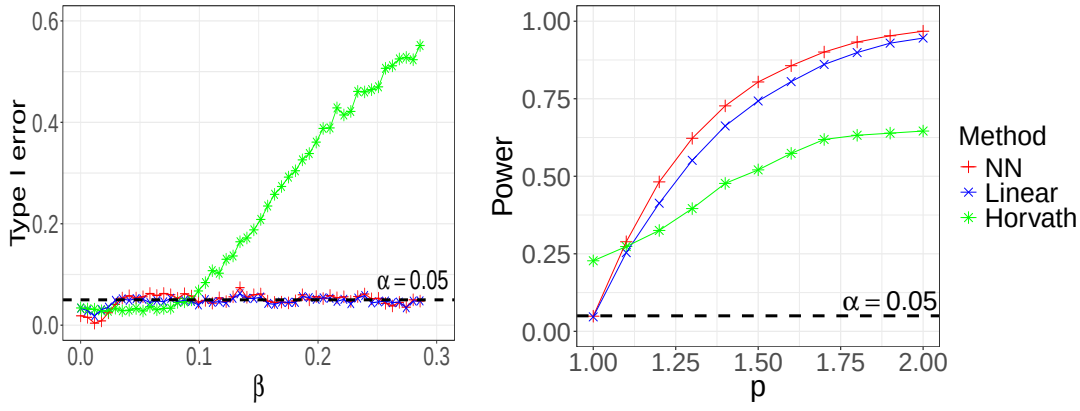
(a) (b)

Figure S.6: Panel (a): Type I errors for neural network (red), linear classifier (blue), and Horváth et al. (2020)'s method (green) when multivariate t-distributions are used for alternative hypothesis H_1, with the other two parameters of a Matérn covariance function fixed, $\sigma^2 = 1$ and $\nu = 0.5$; Panel (b): Averaged powers across all choices of $\beta \in \mathcal{B}_{\text{test}}$ for neural network (red), linear classifier (blue), and Horváth et al. (2020)'s method (green) given a value of the degree of freedom r on the x-axis with the other two parameters of a Matérn covariance function fixed, $\sigma^2 = 1$ and $\nu = 0.5$

.

After conducting some exploratory data analysis, it becomes apparent that the generated samples from multivariate t-distributions exhibit higher sample variances. Therefore, we replace the sample skewness with sample variance among the original six inputs of neural networks. In practical implementations, neural networks can be trained using additional inputs that differentiate normal distributions from other types of distributions. Panel (a) of Figure S.6 displays Type I error rates as the value of β changes. Panel (b) of Figures S.6 displays averaged powers across all βs in $\mathcal{B}_{\text{test}}$ as the degree of freedom of multivariate t-distribution changes, where the infinite value of the degree of freedom denotes a normal distribution. Our adaptive cut-off methods continue to maintain approximate 5% Type I error rates across all values of β when we use t-distributed non-normal data for the alternative hypothesis. Regarding power comparison, neural networks exhibit the highest power despite the method in Horváth et al. (2020) showing inflated Type I error rates.

To demonstrate the applicability of our method to different dependence structures, we consider two additional covariance models for generating spatially dependent normal data, the squared exponential (Gaussian) and spherical covariance models. First, for any two observations $Y(\mathbf{s}_i), Y(\mathbf{s}_j)$ at two generic locations $\mathbf{s}_i, \mathbf{s}_j \in \mathbb{R}^2$, the squared exponential covariance function is:



(a) (b)

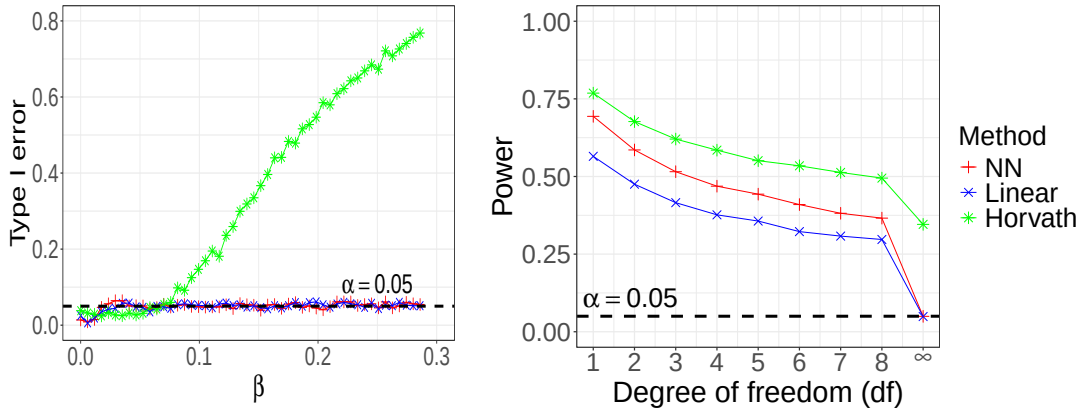
Figure S.7: Panel (a): Type I errors for neural network (red), linear classifier (blue), and Horváth et al. (2020)'s method (green) when signed power transformations are used for alternative hypothesis H_1, with the other parameter of a squared exponential covariance function fixed, $\sigma^2 = 1$; Panel (b): Averaged powers across all choices of $\beta \in \mathcal{B}_{\text{test}}$ for neural network (red), linear classifier (blue), and Horváth et al. (2020)'s method (green) given a value of p on the x-axis with the other parameter of a squared exponential covariance function fixed, $\sigma^2 = 1$.

$$\text{cov}\{Y(\mathbf{s}_i), Y(\mathbf{s}_j)\} = \sigma^2 \exp\left(-\|\boldsymbol{s}_i - \boldsymbol{s}_j\|^2/(2\beta^2)\right),$$

where the parameter σ^2 specifies marginal variance and $\beta > 0$ controls the range of spatial dependence. We fix $\sigma^2 = 1$ and consider varying βs. In the training set, we generate spatially dependent normal data with equally spaced $\beta \in \mathcal{B}_{\text{train}} = \{\beta_1, \ldots, \beta_{\max}\}$ such that $|\mathcal{B}_{\text{train}}| = n_{\beta,\text{train}} = 30$, where $\beta_{\max} = 0.286$ which corresponds to the effective range of 0.7 on a unit square. In the testing set, we generate normal data in a similar manner using equally spaced β from $\mathcal{B}_{\text{test}}$ with $|\mathcal{B}_{\text{test}}| = n_{\beta,\text{test}} = 50$ for which we can evaluate the interpolation ability of neural networks. For non-normal data, we consider the signed power transformation with the same values of p in Section 3 of the main manuscript and multivariate t-distribution with the same values of r in the previous section of this supplement. Figure S.7 and S.8 show the resulting Type I errors and powers when we use the signed power transformations and multivariate t-distributions, respectively.

As another option of a covariance function, we consider spherical models defined as:

$$\text{cov}\{Y(\mathbf{s}_i), Y(\mathbf{s}_j)\} = \sigma^2 \left(1 - \frac{3}{2}(\|\boldsymbol{s}_i - \boldsymbol{s}_j\|/\beta) + \frac{1}{2}(\|\boldsymbol{s}_i - \boldsymbol{s}_j\|/\beta)^3\right) \mathbf{1}_{(\|\boldsymbol{s}_i - \boldsymbol{s}_j\| \leq \beta)},$$



(a) (b)

Figure S.8: Panel (a): Type I errors for neural network (red), linear classifier (blue), and Horváth et al. (2020)'s method (green) when multivariate t-distributions are used for alternative hypothesis H_1, with the other parameter of a squared exponential covariance function fixed, $\sigma^2 = 1$; Panel (b): Averaged powers across all choices of $\beta \in \mathcal{B}_{\text{test}}$ for neural network (red), linear classifier (blue), and Horváth et al. (2020)'s method (green) given a value of degree of freedom on the x-axis with the other parameter of a squared exponential covariance function fixed, $\sigma^2 = 1$

.

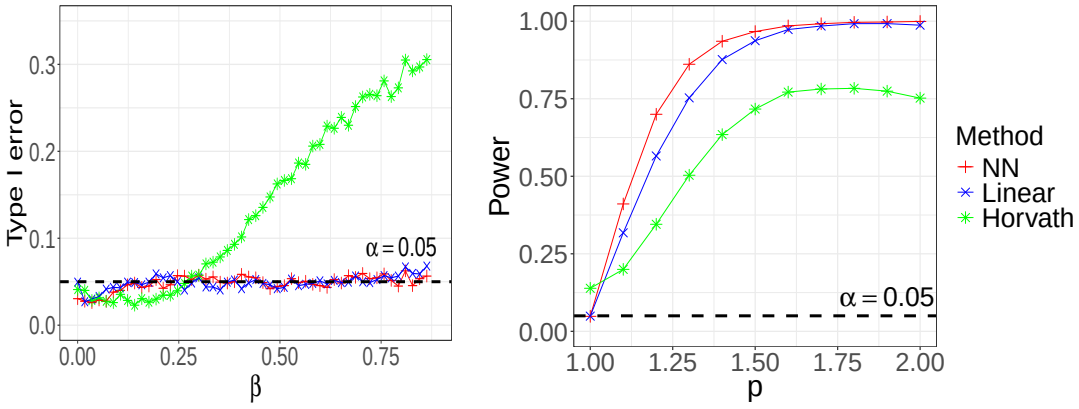
where σ^2 is a variance parameter, while $\beta > 0$ is a parameter which controls the speed of decay of the spatial dependence as the distance between two locations increases. We, again, fix $\sigma^2 = 1$ and consider varying βs. The other settings are the same as in the case of squared exponential covariance, except only for the different $\beta_{\max}$. The value of $\beta_{\max}$ that results in an effective range of 0.7 on a unit square is 0.863. Figure S.9 and S.10 show the resulting Type I errors and powers when we use the signed power transformations and multivariate t-distributions, respectively.

B Section 4: Application

Moving window mean model

The model for the mean is fitted for each location independently. Figure S.11 displays a map (panel (a)) and histogram (b) describing the resulting R^2. The map contains the location with the maximum R^2 marked as a blue cross and the location with the minimum R^2 marked as a white circle. The actual time series at the locations of the blue cross and white circle are shown in Figure S.12.

We also examine the temporal auto-correlations among $\widehat{\eta}_t(\mathbf{s}_i)$ at each



(a) (b)

Figure S.9: Panel (a): Type I errors for neural network (red), linear classifier (blue), and Horváth et al. (2020)'s method (green) when signed power transformations are used for alternative hypothesis H_1, with the other parameter of a spherical covariance function fixed, $\sigma^2 = 1$; Panel (b): Averaged powers across all choices of $\beta \in \mathcal{B}_{\text{test}}$ for neural network (red), linear classifier (blue), and Horváth et al. (2020)'s method (green) given a value of p on the x-axis with the other parameter of a spherical covariance function fixed, $\sigma^2 = 1$

.

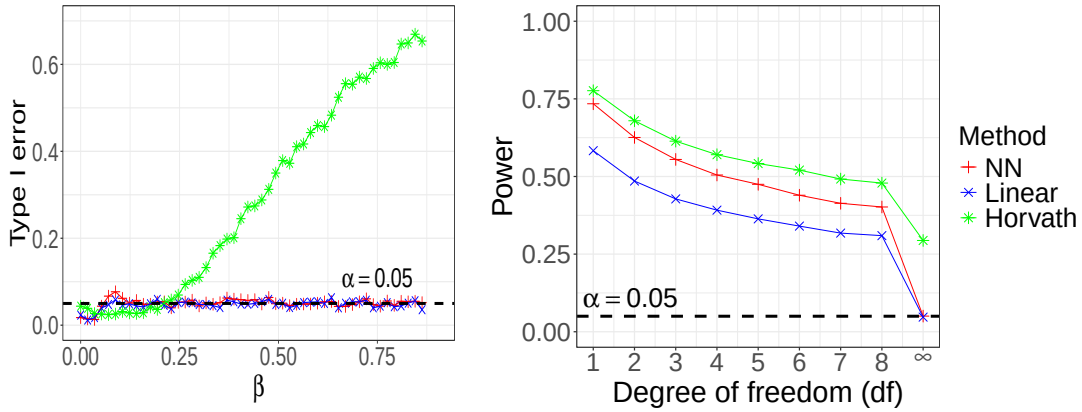
location $i = 1, \ldots, M$. The mean auto-correlation is 0.003 with standard deviation 0.028, which implies temporal independence. Figure S.13 presents the histogram of the auto-correlations across all locations.

Harmonic mean model

As an alternative model, one can consider a harmonic regression model on the mean:

$$Y_t(\mathbf{s}_i) = b_0(\mathbf{s}_i) + \sum_{k=1}^{K} \left\{ b_{k,\sin}(\mathbf{s}_i) \sin\left(\frac{2\pi t}{P_k}\right) + b_{k,\cos}(\mathbf{s}_i) \cos\left(\frac{2\pi t}{P_k}\right) \right\} + \epsilon_t(\mathbf{s}_i) \tag{S.1}$$

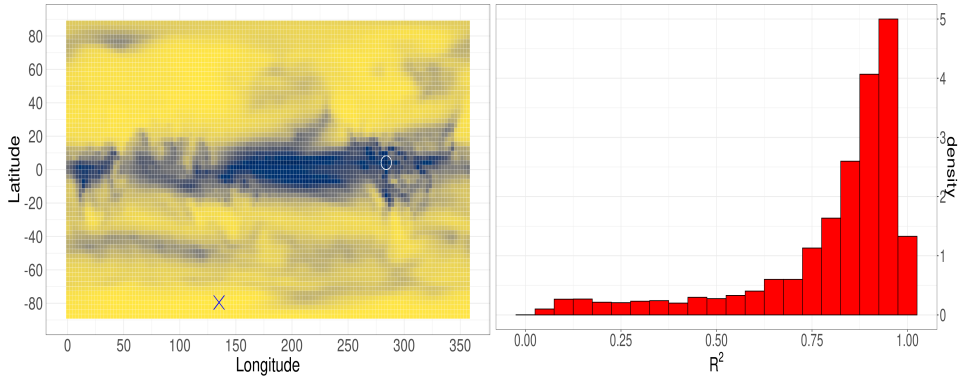
We use $(P_1, P_2, P_3, P_4, P_5) = (12, 36, 72, 108, 144)$ and compute the optimal K for each location based on Bayesian information criterion (BIC). Tabel S.1 summarized the selected optimal K for every location. The average R^2 across all locations is 0.76 with standard deviation 0.24 and 86% values of R^2 are greater than 0.5.



(a) (b)

Figure S.10: Panel (a): Type I errors for neural network (red), linear classifier (blue), and Horváth et al. (2020)'s method (green) when multivariate t-distributions are used for alternative hypothesis H_1, with the other parameter of a spherical covariance function fixed, $\sigma^2 = 1$; Panel (b): Averaged powers across all choices of $\beta \in \mathcal{B}_{\text{test}}$ for neural network (red), linear classifier (blue), and Horváth et al. (2020)'s method (green) given a value of degree of freedom on the x-axis with the other parameter of a spherical covariance function fixed, $\sigma^2 = 1$

.

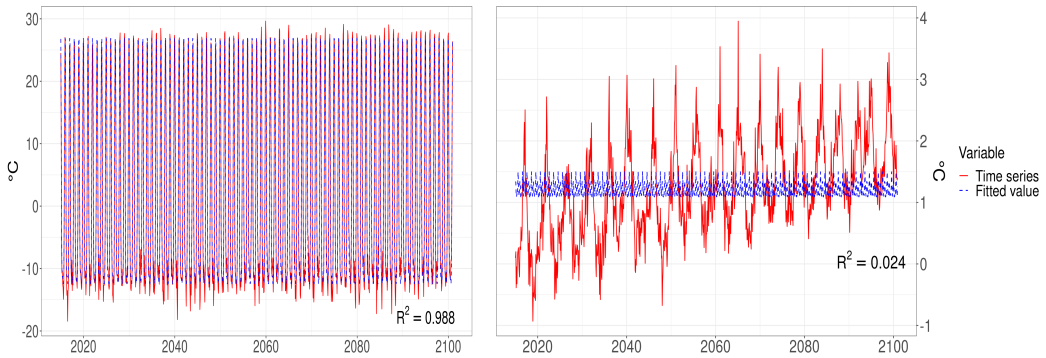


(a) (b)

Figure S.11: Panel (a): Map of fitted R^2 for each location where a blue cross and red cross represent the highest and lowest R^2 respectively; Panel (b): Histogram of R^2 across all locations;

Tabulated range maximum bound

The values of the range maximum bound $\beta_{\max}$ as a function of the smoothness parameter ν and across different levels of aggregation are shown in



(a) (b)

Figure S.12: Panel (a): Time series (red solid line) and fitted values (blue dashed line) at the location of the blue cross with $R^2 = 0.988$; Panel (b): Same as (a) at the location of the white circle with $R^2 = 0.024$. The y-axis on both panels represents temperature anomaly, a departure from a reference value which is the average temperature of the first year (2015).

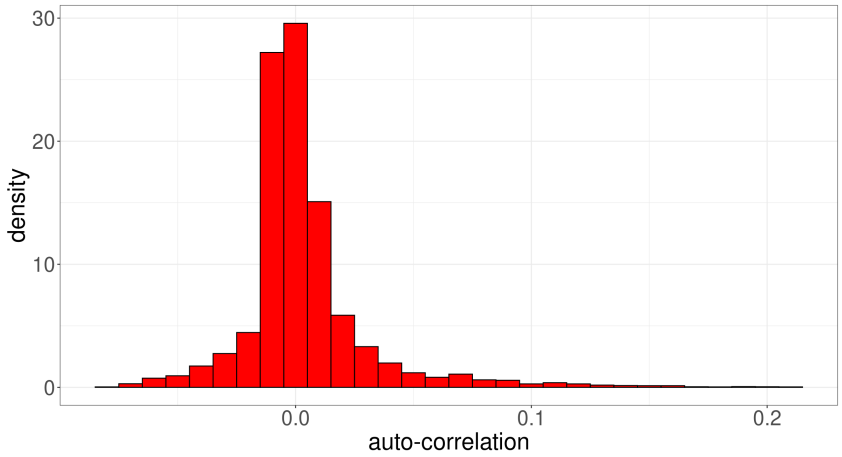


Figure S.13: Auto-correlation histogram of $\widehat{\eta}_t(\mathbf{s}_i)$ across all locations.

Table S.1: Number of locations where the optimal K in (S.1) for each location is selected based on BIC.

Optimal K	1	2	3	4	5
# of locations	7448	86	369	60	229

Table S.2.

Table S.2: Range parameter bound $\beta_{\max}$ as a function of the smoothness parameter ν and across different levels of aggregation.

Smoothness parameter	$\nu = 0.5$	$\nu = 1.0$	$\nu = 1.5$	$\nu = 2.0$	$\nu = 2.5$	$\nu = 3.0$
$\beta_{\max}$	2105.3	1577.3	1417.4	1174.8	944.0	758.2

References

Abdulah, S., Li, Y., Cao, J., Ltaief, H., Keyes, D. E., Genton, M. G., and Sun, Y. (2023). Large-scale environmental data science with ExaGeoStatR. *Environmetrics*, 34(1):Paper No. e2770, 28 p.

Horváth, L., Kokoszka, P., and Wang, S. (2020). Testing normality of data on a multivariate grid. *Journal of Multivariate Analysis*, 179:104640.