ORIGINAL PAPER

A Generalized Unified Skew-Normal Process with Neural Bayes Inference

Kesen Wang[1] · Marc G. Genton[1]

Abstract

In recent decades, statisticians have been increasingly encountering spatial data that exhibit non-Gaussian behaviors such as asymmetry and heavy-tailedness. As a result, the assumptions of symmetry and fixed tail weight in Gaussian processes have become restrictive and may fail to capture the intrinsic properties of the data. To address the limitations of the Gaussian models, a variety of skewed models has been proposed, of which the popularity has grown rapidly. These skewed models introduce parameters that govern skewness and tail weight. Among various proposals in the literature, unified skewed distributions, such as the Unified Skew-Normal (SUN), have received considerable attention. In this work, we revisit a more concise and intepretable re-parameterization of the SUN distribution and apply the distribution to random fields by constructing a generalized unified skew-normal (GSUN) spatial process. We demonstrate that the GSUN is a valid spatial process by showing its vanishing correlation at large distances and provide the corresponding spatial interpolation method. In addition, we develop an inference mechanism for the GSUN process using the concept of neural Bayes estimators with deep graphical attention networks (GATs) and encoder transformer. We show the superiority of our proposed estimator over conventional CNN-based architectures in terms of stability and accuracy by means of a simulation study. In addition, we demonstrate that the GSUN process offers enhanced flexibility compared to another model proposed in the literature through an application to Pb-contaminated soil data. Furthermore, we show that the GSUN process is different from the conventional Gaussian processes and Tukey g-and-h processes, through the probability integral transform (PIT).

Keywords Encoder transformer · Graphical attention network · Neural Bayes estimator · Non-Gaussian process · Unified skew-normal distribution

1 Introduction

In the past decade, there has been a growing interest in applying the Skew-Normal (SN) distribution to model a spatial field due to the skewness parameter that can capture asymmetric data. In particular, Kim and Mallick (2004) constructed a Bayesian predictor based on the SN distribution. In addition, Dominguez-Molina et al. (2003), Allard and Naveau (2007), and Rimstad and Omre (2014) proposed to use the Closed Skew-Normal (CSN), a generalization of the SN distribution, to model skewed geostatistical data

with various methods such as MCMC and the method of moments presented for parameter estimation. Moreover, Márquez-Urbina and González-Farías (2022) proposed a flexible special case of the CSN for spatial modeling and prediction. Despite these fruitful studies, Arellano-Valle and Azzalini (2006) mentioned the non-identifiability and over-parameterization issue within the CSN distribution. Moreover, Genton and Zhang (2012) conducted a comprehensive study on the non-identifiability problem of the SN spatial process and extended the conclusion to elliptically contoured random fields. Furthermore, Minozzo and Ferracuti (2012) demonstrated that SN-type spatial processes do not satisfy the requirement of vanishing correlations at large distances. A possible remedy was developed by Zhang and El-Shaarawi (2010), where a stationary process was constructed with SN as marginal distributions.

To allow the spatial model to capture asymmetry and heavy tails, Tagle et al. (2019) applied the multivariate Skew-

✉ Kesen Wang
  Kesen.Wang@alumni.kaust.edu.sa

  Marc G. Genton
  marc.genton@kaust.edu.sa

[1] Statistics Program, King Abdullah University of Science and Technology, Thuwal 23955-6900, Saudi Arabia

t (ST) distribution to model daily wind data. Moreover, Bevilacqua et al. (2021) developed a ST process model that addresses the regression and dependence analysis of asymmetric and heavy-tailed spatial data. In addition, Alodat and Al-Rawwash (2014) , Alodat and Shakhatreh (2020), and Benavoli et al. (2020) have also tackled the regression problem of asymmetric data by proposing an extended skew-Gaussian process for regression and a Gaussian process regression with skewed errors.

In addition to SN or ST spatial models, practitioners also sought to construct spatial models using the Unified Skew-Normal (SUN) distribution. To this end, a comprehensive summary of the existence of the SN and SUN family-based random fields can be found in Mahmoudian (2018). A particularly noteworthy SUN-based spatial process is the Unified Skew Gaussian-Log-Gaussian (SUGLG) process proposed in Zareifard and Khaledi (2013) with a Stochastic Approximation Expectation Maximization (SAEM) algorithm for parameter inference. The SUGLG process is a more complicated model than the Gaussian process and can be used to model the asymmetric property of the spatial data, and the SAEM algorithm was proposed to avoid the intractable computation of the integrals involved in the log-likelihood function. In addition, correlations vanish at large distances in the SUGLG process because it is a convolution of a Gaussian process (dominant) and a form of truncated Gaussian process (latent), where both processes follow such a property. Although the SUGLG can address numerous challenges existing in the SN and ST processes, the SAEM is likely to be unstable and time-consuming with large spatial dimensions. The simulation study and the real data application conducted in Zareifard and Khaledi (2013) were limited to sample sizes of about 100. Moreover, the SUGLG model is restricted to a simplified sub-model of the SUN discussed in Wang et al. (2023b), which imposes the same dependence structure on the covariance of the observed and latent processes and the skewness matrix. Hence, in this work, we generalize the SUGLG and propose the Generalized Unified Skew-Normal (GSUN) process by involving a separate dependence structure also in the latent dimensions and designing a principled diagonal structure on the skewness matrix to improve its statistical interpretability and maintain vanishing correlations at large distances of the process. The skewness parameter accounts for information from both the observed and latent processes instead of the observed alone. Furthermore, we adopt a re-parameterization of the SUN distribution; the new parameterization is numerically more stable and geometrically more intuitive compared with the original parameterization proposed in Arellano-Valle and Azzalini (2006).

Sainsbury-Dale et al. (2025) demonstrated that a Graph Neural Network (GNN) can accurately approximate Bayes estimators, a function that maps the realizations of spatial processes to the point estimates of the corresponding parameters, which is amortized and efficient in inference. The idea of using neural networks to map realizations to the point estimates of the parameters is defined as the neural Bayes estimator (Sainsbury-Dale et al. 2024). Furthermore, Lenzi et al. (2023), Richards et al. (2024), and Walchessen et al. (2024) have applied variants of the neural Bayes estimators to infer complex processes and distributions, for which likelihood functions are intractable or computationally intensive. A comprehensive review of neural Bayes estimators and amortized parameter inference can be found in Zammit-Mangion et al. (2025). Following this direction, we combine a deep Graph Attention Network (GAT) (Veličković et al. 2018) with a multi-layer encoder transformer architecture (Vaswani 2017) to construct the neural Bayes estimator for the GSUN model, which is adapted to regularly and irregularly spaced grids. Moreover, we develop a simulation-based uncertainty quantification method for the parameter estimates.

This paper is organized as follows. Section 2 revisits the SUN distribution and the more concise re-parameterization, laying the ground for the proposition of the GSUN model. Section 3 formally introduces the GSUN spatial process including its definition, validity, and spatial interpolation. Section 4 gives a comprehensive introduction to the neural Bayes estimator. Section 5 presents the network architecture and data representation methods adopted in this work. Section 6 conducts a series of comparative and evaluative simulation studies of the GSUN process to demonstrate its efficiency. Section 7 applies the GSUN model to a dataset of Pb-contaminated soils and draws a comparison between the GSUN and SUGLG model. Section 8 concludes and outlines the contributions of this work.

2 Revisiting the SUN Distribution

In this section, we briefly revisit the Unified Skew-Normal (SUN) distribution proposed in Arellano-Valle and Azzalini (2006) and introduce a more efficient parameterization documented in Arellano-Valle and Genton (2026).

2.1 The SUN Distribution

As previously mentioned, to address the non-identifiability issue existing in the CSN distribution, Arellano-Valle and Azzalini (2006) proposed a unified framework involving many skewed models as sub-cases by enforcing $\bar{\Omega}^*$ below to be a correlation matrix. The SUN distribution arises from the selection mechanism described in Arellano-Valle and Azzalini (2006), starting from a joint multivariate Gaussian

random vector

$$\mathbf{U} = \begin{pmatrix} \mathbf{U}_0 \\ \mathbf{U}_1 \end{pmatrix} \sim \mathcal{N}_{m+d}(\mathbf{0}, \bar{\mathbf{\Omega}}^*), \qquad \bar{\mathbf{\Omega}}^* = \begin{pmatrix} \bar{\mathbf{\Gamma}} & \mathbf{\Delta}^\top \\ \mathbf{\Delta} & \bar{\mathbf{\Omega}} \end{pmatrix},$$

where $\bar{\mathbf{\Gamma}}$ is the latent correlation matrix, $\mathbf{\Delta}$ represents the skewness matrix, and $\bar{\mathbf{\Omega}} = \omega^{-1}\mathbf{\Omega}\omega^{-1}$ defines the correlation matrix, with $\omega = \mathrm{diag}(\mathbf{\Omega})^{1/2}$. Then a random vector $\mathbf{Z} = (\mathbf{U}_1|\mathbf{U}_0 + \tau > \mathbf{0})$, where $\tau \in \mathbb{R}^m$ is a truncation parameter, is a non-shifted SUN random vector. With a location shift parameter $\xi \in \mathbb{R}^d$, $\mathbf{Y} = \xi + \omega\mathbf{Z}$ is a fully parameterized SUN random vector. The probability density function (pdf) of $\mathbf{Y}$ is

$$f(\mathbf{y}) = \phi_d(\mathbf{y} - \xi; \mathbf{\Omega})$$
$$\frac{\Phi_m(\tau + \mathbf{\Delta}^\top \bar{\mathbf{\Omega}}^{-1}\omega^{-1}(\mathbf{y} - \xi); \bar{\mathbf{\Gamma}} - \mathbf{\Delta}^\top\bar{\mathbf{\Omega}}^{-1}\mathbf{\Delta})}{\Phi_m(\tau; \bar{\mathbf{\Gamma}})}, \quad \mathbf{y} \in \mathbb{R}^d.$$
$$(1)$$

In this work, we denote the SUN random vector with pdf defined in (1) as $\mathcal{SUN}_{d,m}(\xi, \mathbf{\Omega}, \mathbf{\Delta}, \tau, \bar{\mathbf{\Gamma}})$. Numerous interesting and practical properties, such as moments of the SUN, were studied by Arellano-Valle and Azzalini (2006), Gupta et al. (2013), Arellano-Valle and Azzalini (2022), and Wang et al. (2023b).

2.2 Re-parameterization of the SUN Distribution

In Section 2.1, we have denoted a SUN random vector by $\mathbf{Y} \sim \mathcal{SUN}_{d,m}(\xi, \mathbf{\Omega}, \mathbf{\Delta}, \tau, \bar{\mathbf{\Gamma}})$, which is the original parameterization proposed in Arellano-Valle and Azzalini (2006). Although such a parameterization has many intriguing advantages, the covariance matrix $\mathbf{\Omega}^*$ induces numerical instability in simulating realizations. This numerical instability can be observed from the *rsun* function provided in the *sn* R-package (Azzalini 2023), which simulates replicates of the SUN random vector using exactly this particular parameterization and stochastic representation. The simulations often run into problems with rather large values of the entries in $\mathbf{\Delta}$, causing $\bar{\mathbf{\Omega}}^*$ to be singular and hence to fail in Cholesky factorization, a crucial step to produce the samples. Alternatively, $\mathbf{Y}$ can also be represented with the convolution mechanism proposed in Arellano-Valle and Azzalini (2006). In this sense, the introduction of two intermediate matrices $\mathbf{B}_0$ and $\mathbf{B}_1$ obscures the directions of the skewness of the random vector. In other words, $\mathbf{\Delta}$ introduces the skewness but does not indicate the direction, making $\mathbf{\Delta}$ less interpretable. Meanwhile, this parameterization complicates simulations of random samples from the convolution mechanism.

To address these disadvantages, Arellano-Valle and Genton (2026) provided a simple way to construct a random vector with SUN distribution. In detail, they let $\omega\mathbf{\Delta} = \mathbf{H}\bar{\mathbf{\Gamma}}$

and $\mathbf{\Omega} = \mathbf{\Psi} + \mathbf{H}\bar{\mathbf{\Gamma}}\mathbf{H}^\top$. Then, they re-parameterized the definition of unified skew-normal distribution in Arellano-Valle and Azzalini (2022) using the convolution mechanism proposed in Arellano-Valle and Azzalini (2006) and denoted it as $\mathcal{SUN}_{d,m}(\xi, \mathbf{\Psi}, \mathbf{H}, \tau, \bar{\mathbf{\Gamma}})$. Here is a formal definition:

Definition 1 A d-dimensional random vector $\mathbf{Y}$ has a unified skew-normal distribution, denoted as $\mathbf{Y} \sim \mathcal{SUN}_{d,m}(\xi, \mathbf{\Psi}, \mathbf{H}, \tau, \bar{\mathbf{\Gamma}})$, with location parameter $\xi \in \mathbb{R}^p$, dispersion matrix $\mathbf{\Psi} \in \mathbb{R}^{d \times d}$, shape/skewness parameters $\mathbf{H} \in \mathbb{R}^{d \times m}$, latent truncation parameter $\tau \in \mathbb{R}^m$, and positive definite latent correlation matrix $\bar{\mathbf{\Gamma}} \in \mathbb{R}^{m \times m}$, if $\mathbf{Y} = \xi + \mathbf{H}\mathbf{U} + \mathbf{W}$, where $\mathbf{U} \overset{d}{=} (\mathbf{W}_0|\mathbf{W}_0 + \tau > \mathbf{0})$, with $\mathbf{W}_0 \sim \mathcal{N}_m(\mathbf{0}, \bar{\mathbf{\Gamma}})$ and $\mathbf{W} \sim \mathcal{N}_d(\mathbf{0}, \mathbf{\Psi})$ having independent, multivariate, Gaussian distributions.

The new parameterization leads to the same distribution as with the original, see Section S.1 in the Supplementary Materials for a detailed proof. Moreover, the new parameterization is numerically stable and eliminates the issues of Cholesky factorization in the original parameterization. In addition, the new skewness parameter $\mathbf{H}$ provides a direct representation of the direction of the skewed mass. The convolution representation $\mathbf{Y} = \xi + \mathbf{H}\mathbf{U} + \mathbf{W}$ can be understood as a symmetric normal random vector $\mathbf{W}$ plus a truncated normal random vector $\mathbf{U}$, with $\mathbf{H}$ determining the asymmetry introduced by $\mathbf{U}$ and controlling the direction.

Lastly, the re-parameterization also allows for a more straightforward computation of the mean and variance. Because $\mathbf{Y} = \xi + \mathbf{H}\mathbf{U} + \mathbf{W}$, then $\mathbb{E}(\mathbf{Y}) = \xi + \mathbf{H}\mathbb{E}(\mathbf{U})$ and $\mathrm{var}(\mathbf{Y}) = \mathbf{H}\mathrm{var}(\mathbf{U})\mathbf{H}^\top + \mathbf{\Psi}$. Note that the mean and variance of multivariate truncated normal random vectors can be computed numerically with the *R*-package *tmvtnorm* (Wilhelm and Manjunath 2023).

3 The GSUN Spatial Model

In this section, we study the GSUN spatial model by first defining the GSUN spatial process and then deriving the kriging formula for spatial interpolation.

3.1 GSUN Spatial Processes

Zhang and El-Shaarawi (2010) discussed how the SN spatial model faces a non-ergodic issue, which is crucial for a valid and reasonable spatial process. The non-ergodic issue can also be generalized to the ST spatial model. The problem that causes SN and ST to be non-ergodic is that all locations in the study region share an identical univariate truncated normal or truncated t random variable. As a result, the covariance or correlation between two locations persists even if they are significantly distant from each other.

To address these limitations, we propose a more flexible construction of the SUN spatial model that simultaneously satisfies the requirement of vanishing correlations at large distances and accounts for a latent spatial process of skewness. Specifically, we define a GSUN spatial process as

$$Z(\mathbf{s}) = W(\mathbf{s}) + h(\mathbf{s})W^+(\mathbf{s}),$$

where $W(\mathbf{s})$ and $W^+(\mathbf{s}) \stackrel{d}{=} (U(\mathbf{s}) \mid U(\mathbf{s}) > 0)$, with $U(\mathbf{s})$ being a zero-mean Gaussian process, are independent Gaussian and truncated Gaussian spatial processes with Matérn covariance and correlation functions (see Wang et al. (2023a) for the choice of parameterization), respectively. We adopt the Matérn covariance rather than the RBF kernel because it offers greater flexibility and interpretability, with a smoothness parameter ν controlling mean-square differentiability; the RBF kernel is the limiting case $\nu \to \infty$, yielding often unrealistically smooth sample paths for geo-spatial applications. Here $h(\mathbf{s})$, a location-specific scalar that dictates the amount of the truncated Gaussian process $W^+(\mathbf{s})$ added to $W(\mathbf{s})$, can be specified with any legitimate real values.

If observed at n locations, then let $\mathbf{Z} = \{Z(\mathbf{s}_1), \ldots, Z(\mathbf{s}_n)\}^\top$, $\mathbf{W} = \{W(\mathbf{s}_1), \ldots, W(\mathbf{s}_n)\}^\top$, and $\mathbf{W}^+ = \{W^+(\mathbf{s}_1), \ldots, W^+(\mathbf{s}_n)\}^\top$. We have $\mathbf{W} \sim \mathcal{N}_n(\mathbf{0}, \boldsymbol{\Sigma}(\boldsymbol{\theta}_1))$ and $\mathbf{W}^+ \sim \mathcal{TN}_n(\mathbf{0}; \mathbf{0}, \mathbf{C}(\boldsymbol{\theta}_2))$, where $\mathcal{TN}_n(\mathbf{a}; \mathbf{m}, \mathbf{K})$ represents an n-dimensional truncated Gaussian distribution cut from below $\mathbf{a}$ with mean vector $\mathbf{m}$ and covariance matrix $\mathbf{K}$; $\boldsymbol{\Sigma}(\boldsymbol{\theta}_1)$ and $\mathbf{C}(\boldsymbol{\theta}_2)$ are the Matérn covariance and correlation matrices with $\boldsymbol{\theta}_1 = (\sigma^2, \beta_1, \nu_1)^\top$ and $\boldsymbol{\theta}_2 = (\beta_2, \nu_2)^\top$.

We then define the skewness matrix as

$$\mathbf{H} = \mathrm{diag}(h(\mathbf{s}_1), \ldots, h(\mathbf{s}_n)) = \delta_1 \mathbf{I}_n$$
$$+ \delta_2 \mathrm{diag}\left\{ \frac{1}{2n} \sum_{i=1}^n \left(\lambda_{i,\boldsymbol{\Sigma}(\boldsymbol{\theta}_1)} \mathbf{P}_{i,\boldsymbol{\Sigma}(\boldsymbol{\theta}_1)} + \lambda_{i,\mathbf{C}(\boldsymbol{\theta}_2)} \mathbf{P}_{i,\mathbf{C}(\boldsymbol{\theta}_2)} \right) \right\},$$

where $(\lambda_{i,\boldsymbol{\Sigma}(\boldsymbol{\theta}_1)}, \mathbf{P}_{i,\boldsymbol{\Sigma}(\boldsymbol{\theta}_1)})$ and $(\lambda_{i,\mathbf{C}(\boldsymbol{\theta}_2)}, \mathbf{P}_{i,\mathbf{C}(\boldsymbol{\theta}_2)})$ denote the i-th eigenvalue-eigenvector pairs of the covariance and correlation matrices of the observed and latent Gaussian processes, respectively. The coefficients $\delta_1 \in \mathbb{R}$ and $\delta_2 \in \mathbb{R}$ control the global and spatially varying skewness effects.

Motivation and interpretation. The construction of $\mathbf{H}$ follows three guiding principles: *interpretability*, *statistical parsimony*, and *ergodicity*. Each diagonal entry $h(\mathbf{s}_i)$ represents the local magnitude of skewness at location $\mathbf{s}_i$, and its definition through the weighted eigen-decomposition of $\boldsymbol{\Sigma}(\boldsymbol{\theta}_1)$ and $\mathbf{C}(\boldsymbol{\theta}_2)$ provides a clear geometric interpretation. The eigenvectors $\mathbf{P}_{i,\boldsymbol{\Sigma}(\boldsymbol{\theta}_1)}$ and $\mathbf{P}_{i,\mathbf{C}(\boldsymbol{\theta}_2)}$ identify the principal directions of spatial dependence, while the corresponding eigenvalues quantify the contribution of each direction. Consequently, $h(\mathbf{s}_i)$ aggregates the dominant modes of spatial variability, assigning larger weights to directions associated with higher eigenvalues, and thereby aligns local skewness

with the main axes of spatial dependence. This coupling ensures that asymmetry in the process reflects intrinsic spatial patterns rather than random perturbations.

Furthermore, the two coefficients (δ_1, δ_2) decompose the overall skewness into a global and a spatially varying component: $\delta_1 \mathbf{I}_n$ captures a shared, direction-invariant asymmetry across the entire domain, while δ_2 scales the contribution of the principal components of the covariance and correlation structures. This linear combination is intentionally parsimonious, analogous to a linear regression decomposition, where δ_1 acts as an intercept representing baseline skewness and δ_2 as a slope parameter modulating the effect of spatial eigenstructure. Such a regression-style design balances flexibility with simplicity, avoiding the over-parameterization that would arise from a full or unconstrained $\mathbf{H}$, while maintaining interpretability and coherence with the spatial correlation geometry.

Under this construction, when $\delta_2 = 0$, the GSUN process becomes stationary because $h(\mathbf{s})$ is constant across locations; otherwise, $\delta_2 \neq 0$ introduces spatially adaptive skewness determined by the leading eigen-directions of $\boldsymbol{\Sigma}(\boldsymbol{\theta}_1)$ and $\mathbf{C}(\boldsymbol{\theta}_2)$, producing a controlled form of non-stationarity. In compact notation, $\mathbf{Z} \sim \mathcal{SUN}_{n,n}(\mathbf{0}, \boldsymbol{\Sigma}(\boldsymbol{\theta}_1), \mathbf{H}, \mathbf{0}, \mathbf{C}(\boldsymbol{\theta}_2))$, and equivalently $\mathbf{Z} = \mathbf{W} + \mathbf{H}\mathbf{W}^+$.

Compared with the SUN spatial model in Zareifard and Khaledi (2013), the GSUN process allows for a distinct latent dependence structure with its own range and smoothness parameters, thereby extending the flexibility of dependence modeling in the skewed component. Defining $\mathbf{H}$ as a diagonal matrix composed of a linear combination of a global skewness term and a weighted average of eigenvectors leads to a more interpretable and stable representation than scaling $\boldsymbol{\Sigma}(\boldsymbol{\theta}_1)$ by a constant, as done previously.

Ergodicity and smoothness. The GSUN process also satisfies the requirement of vanishing correlations at large distances. Specifically,

$$\mathrm{cov}\{Z(\mathbf{s}_i), Z(\mathbf{s}_j)\} = \boldsymbol{\Sigma}(\boldsymbol{\theta}_1)_{i,j} + \mathbf{H}_{i,i} \, \mathrm{var}(\mathbf{W}^+)_{i,j} \mathbf{H}_{j,j}.$$

As the distance between $\mathbf{s}_i$ and $\mathbf{s}_j$ grows, both $\boldsymbol{\Sigma}(\boldsymbol{\theta}_1)_{i,j}$ and $\mathrm{var}(\mathbf{W}^+)_{i,j}$ converge to zero under the Matérn kernel and Proposition 2 in the Supplementary Materials, implying $\mathrm{cov}\{Z(\mathbf{s}_i), Z(\mathbf{s}_j)\} \approx 0$. Thus, the process is ergodic, in contrast to the classical SN and ST spatial processes.

Finally, regarding mean-square differentiability, when a spatial process is a convolution of a smooth and a rough process, the differentiability is inherited from the rougher component. Hence, for GSUN, the mean-square differentiability depends primarily on ν_1 or ν_2, whichever is smaller.

3.2 Spatial Interpolation

We can rely on the kriging method for spatial interpolation. For instance, we note that the SUN has closed conditional distributions (Arellano-Valle and Azzalini 2022; Wang et al. 2024). Let

$$
\begin{pmatrix} \mathbf{Z}_1 \\ \mathbf{Z}_2 \end{pmatrix} \sim \mathcal{SUN}_{n_1+n_2,n} \left(\begin{pmatrix} \boldsymbol{\xi}_1 \\ \boldsymbol{\xi}_2 \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Psi}_{11} & \boldsymbol{\Psi}_{12} \\ \boldsymbol{\Psi}_{21} & \boldsymbol{\Psi}_{22} \end{pmatrix}, \begin{pmatrix} \mathbf{H}_1 \\ \mathbf{H}_2 \end{pmatrix}, \boldsymbol{\tau}, \bar{\boldsymbol{\Gamma}} \right)
$$

with $n = n_1 + n_2$. Then, following from Proposition 3.2 in Arellano-Valle and Genton (2010) with a Gaussian density generator, we can plug in the new parameterization proposed in Section 2.2 to obtain the conditional distribution of the SUN as $(\mathbf{Z}_1|\mathbf{Z}_2 = \mathbf{z}_2) \sim \mathcal{SUN}_{n_1,n}(\boldsymbol{\xi}_{1\cdot 2}, \boldsymbol{\Psi}_{1\cdot 2}, \mathbf{H}_{1\cdot 2}, \boldsymbol{\tau}_{1\cdot 2}, \bar{\boldsymbol{\Gamma}}_{1\cdot 2})$, where

$$
\begin{aligned}
\boldsymbol{\xi}_{1\cdot 2} &= \boldsymbol{\xi}_1 + \left(\boldsymbol{\Psi}_{12}\boldsymbol{\Psi}_{22}^{-1} + \mathbf{H}_{1\cdot 2}\bar{\boldsymbol{\Gamma}}_{1\cdot 2}\mathbf{H}_{1\cdot 2}^{\top}\boldsymbol{\Psi}_{22}^{-1} \right)(\mathbf{z}_2 - \boldsymbol{\xi}_2), \\
\mathbf{H}_{1\cdot 2} &= (\mathbf{H}_1 - \boldsymbol{\Psi}_{12}\boldsymbol{\Psi}_{22}^{-1}\mathbf{H}_2)\boldsymbol{\gamma}_{1\cdot 2}, \\
\boldsymbol{\Psi}_{1\cdot 2} &= \boldsymbol{\Psi}_{11} - \boldsymbol{\Psi}_{12}\boldsymbol{\Psi}_{22}^{-1}\boldsymbol{\Psi}_{21}, \\
\boldsymbol{\tau}_{1\cdot 2} &= \boldsymbol{\gamma}_{1\cdot 2}^{-1}\{\boldsymbol{\tau} + \boldsymbol{\Gamma}_{1\cdot 2}\mathbf{H}_2^{\top}\boldsymbol{\Psi}_{22}^{-1}(\mathbf{z}_2 - \boldsymbol{\xi}_2)\}, \\
\bar{\boldsymbol{\Gamma}}_{1\cdot 2} &= \boldsymbol{\gamma}_{1\cdot 2}^{-1}(\bar{\boldsymbol{\Gamma}}^{-1} + \mathbf{H}_2^{\top}\boldsymbol{\Psi}_{22}^{-1}\mathbf{H}_2)^{-1}\boldsymbol{\gamma}_{1\cdot 2}^{-1}, \\
\boldsymbol{\gamma}_{1\cdot 2} &= \mathrm{diag}\{(\bar{\boldsymbol{\Gamma}}^{-1} + \mathbf{H}_2^{\top}\boldsymbol{\Psi}_{22}^{-1}\mathbf{H}_2)^{-1}\}^{1/2}.
\end{aligned}
$$

Here, $\mathbf{Z}_1$ represents the values to be predicted, and $\mathbf{z}_2$ represents the observed values. Interpolated values can be obtained by calculating the first moment of $\mathbf{Z}_1|\mathbf{Z}_2$. Additionally, we can determine location-specific prediction intervals by computing the covariance matrix of $\mathbf{Z}_1|\mathbf{Z}_2$ and extracting the marginal variances from the diagonal. Both computations are efficient and straightforward as described in Section 2.2. In addition, the computations of conditional distribution and expectation are implemented in the *sn R*-package (Azzalini 2023).

4 Neural Bayes Estimators

4.1 Background

Parameter inference for high-dimensional spatial models has long been challenging due to the significant computational costs involved in evaluating high-dimensional multivariate probability and cumulative distribution functions. The GSUN model, like others, requires the assessment of both multivariate Gaussian density and the ratios of two multivariate cumulative Gaussian distributions. Deep learning methods have become more popular as a potential solution to mitigate the heavy computation costs associated with distribution functions. In brief, these approaches mainly focus

on approximating the likelihood function, the likelihood-to-evidence ratio, and the posterior distribution using deep neural networks (Sainsbury-Dale et al. 2025). Furthermore, Zammit-Mangion et al. (2025) provided a detailed and comprehensive discussion of the theoretical basis and applications of deep learning methodologies in statistical inference and modeling.

Recent advances in deep learning have led to the development of a new parameter point estimator called the neural Bayes estimator. This estimator uses deep neural networks to map a set of data samples to estimates of the underlying parameters. As described in the study by Sainsbury-Dale et al. (2025), the neural Bayes estimator is computationally efficient, approximately Bayes, likelihood-free, and amortized. Once trained on sufficient data, it allows for much faster inference compared to traditional methods like Markov Chain Monte Carlo (MCMC) or maximum likelihood estimation. The neural Bayes estimator was initially used in spatial statistics with deep Convolutional Neural Networks (CNNs) introduced in Chapter 9 of Goodfellow et al. (2016). However, this approach was limited to spatial data on regularly spaced grids. Sainsbury-Dale et al. (2025) extended the capabilities of the neural Bayes estimator to work with irregularly spaced grids using graphical neural networks, greatly expanding its potential applications.

4.2 Link to Bayes Risk

In this section, we extend the concept of the neural Bayes estimator to the proposed GSUN process. The starting point is the theoretical Bayes risk, defined as the posterior expected loss:

$$
\mathcal{R}(\boldsymbol{\beta}) = \int_{\boldsymbol{\Theta}}\int_{\mathbf{Z}} L\left(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}}_{(\mathbf{Z},\boldsymbol{\beta})} \right) f(\mathbf{Z}|\boldsymbol{\Theta}) \, \mathrm{d}\mathbf{Z} \, \mathrm{d}\Pi(\boldsymbol{\Theta}), \qquad (2)
$$

where $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \delta_1, \delta_2)$ denotes the parameter vector, $\Pi(\cdot)$ is a prior measure on the parameter space, and $f(\mathbf{Z}|\boldsymbol{\Theta})$ is the joint density of the GSUN spatial process conditional on $\boldsymbol{\Theta}$. Specifically, $\mathbf{Z}|\boldsymbol{\Theta} \sim \mathcal{SUN}_{n,n}(\mathbf{0}, \boldsymbol{\Sigma}(\boldsymbol{\theta}_1), \mathbf{H}, \mathbf{0}, \mathbf{C}(\boldsymbol{\theta}_2))$, with $\boldsymbol{\Sigma}(\boldsymbol{\theta}_1)$ and $\mathbf{C}(\boldsymbol{\theta}_2)$ being the Matérn covariance and correlation matrices, respectively.

The Bayes estimator of $\boldsymbol{\Theta}$ is formally defined as

$$
\hat{\boldsymbol{\Theta}}_{(\mathbf{Z})}^{\mathrm{Bayes}} = \arg \min_{\hat{\boldsymbol{\Theta}}} \, \mathbb{E}_{\boldsymbol{\Theta}|\mathbf{Z}}\left[L\left(\boldsymbol{\Theta}, \hat{\boldsymbol{\Theta}} \right) \right],
$$

which explicitly depends on the choice of the loss function $L(\cdot, \cdot)$. For instance, under a quadratic loss, the Bayes estimator corresponds to the posterior mean, whereas under an absolute loss it corresponds to the posterior median. In general, $\hat{\boldsymbol{\Theta}}_{(\mathbf{Z})}^{\mathrm{Bayes}}$ does not admit a closed-form solution due to the high dimensionality of $\boldsymbol{\Theta}$ and the intractability of the SUN posterior.

To approximate (2) in practice, we employ the neural Bayes estimator proposed by Sainsbury-Dale et al. (2025), which minimizes a Monte Carlo approximation of the Bayes risk over simulated data:

$$\hat{\boldsymbol{\beta}} \equiv \arg\min_{\boldsymbol{\beta}} \frac{1}{K} \sum_{k=1}^{K} L\left(\boldsymbol{\Theta}^k, \hat{\boldsymbol{\Theta}}_{(\mathbf{z}_k, \boldsymbol{\beta})}\right), \tag{3}$$

where $\mathbf{z}_k$ is the kth realization of the GSUN process simulated from $f(\mathbf{Z}|\boldsymbol{\Theta}^k)$ and $\hat{\boldsymbol{\Theta}}_{(\mathbf{z}_k, \boldsymbol{\beta})}$ denotes the point estimate of $\boldsymbol{\Theta}$ predicted by a neural network parameterized by $\boldsymbol{\beta}$. The loss function $L(\cdot, \cdot)$ here is the same as in (2), ensuring that the empirical risk in (3) is a consistent Monte Carlo approximation of the theoretical Bayes risk as $K \to \infty$. According to the universal approximation theorem (Hornik et al. 1989; Zhou 2020), a sufficiently large and deep neural network can approximate any measurable function, including the true Bayes estimator $\hat{\boldsymbol{\Theta}}_{(\mathbf{Z})}^{\text{Bayes}}$. Hence, minimizing (3) effectively provides a data-driven, neural approximation to the intractable posterior mapping $\mathbf{Z} \mapsto \hat{\boldsymbol{\Theta}}_{(\mathbf{Z})}^{\text{Bayes}}$.

4.3 Algorithm

The training process of a neural Bayes estimator must involve a large number of simulations, K. Specifically, we start by simulating $\boldsymbol{\Theta}^k$ from its prior and then simulate $\mathbf{Z}^{k,n}$, where n denotes the sample size and k represents the count of training samples, from the corresponding distribution. When choosing priors, Lenzi et al. (2023) advised against using priors with concentrated weights on a particular value, such as the Gaussian prior. Therefore, in our training process, we use uniform priors for all parameters. Additionally, given the need

---

Algorithm 1 Training and Inference of Neural Bayes Estimators

Training

Require: Sample size n, number of replicates N, spatial model $f(\mathbf{Z}|\boldsymbol{\Theta})$, prior $\Pi(\boldsymbol{\Theta})$, neural network architecture for $\hat{\boldsymbol{\Theta}}_{(\mathbf{z}^n, \boldsymbol{\beta})}$, $L(\cdot, \cdot)$ a non-negative loss function, a dropout rate D

Procedure:

(1) Simulate $\boldsymbol{\Theta}^k \sim \Pi(\boldsymbol{\Theta})$

(2) Simulate N replicates of $\mathbf{Z}^{k,n} \sim f(\mathbf{Z}|\boldsymbol{\Theta}^k)$

(3) Compute $\hat{\boldsymbol{\Theta}}_{(\mathbf{z}^{k,n}, \boldsymbol{\beta})}$ with dropout rate D on each layer

(4) Compute $L(\boldsymbol{\Theta}^k, \hat{\boldsymbol{\Theta}}_{(\mathbf{z}^{k,n}, \boldsymbol{\beta})}^k)$

(5) Backward propagate and update $\boldsymbol{\beta}$

(6) Repeat (1) - (5) until a pre-specified criterion is met

Stopping Criterion:

The loss $L(\boldsymbol{\Theta}^k, \hat{\boldsymbol{\Theta}}_{(\mathbf{z}^{k,n}, \boldsymbol{\beta})}) < \epsilon_L$ for L consecutive simulations

Inference

Require: the observed sample $\mathbf{z}$

Procedure:

Plug in the sample $\mathbf{z}$ into the neural networks directly and obtain $\hat{\boldsymbol{\Theta}}_{(\mathbf{z}^n, \boldsymbol{\beta})}$

---

to repeatedly simulate new parameters and data for the neural network, an "on-the-fly" simulation is an advantageous approach to prevent overfitting the network (Sainsbury-Dale et al. 2024) on specific data samples. In other words, for each training step, we simulate $\boldsymbol{\Theta}^k$ and $\mathbf{Z}^{k,n}$ from scratch and then input the data into the proposed network to update $\boldsymbol{\beta}$.

One common challenge in spatial statistics is the limited number of replicates N, which has led to the development of various covariance functions (for more information, see Wang et al. (2023a)). Conventionally, we can assume that $N = 1$; however, based on the empirical evidence from the training stage, it appears that setting N to a slightly larger value, such as 10, can accelerate the learning process. This could be because averaging across repetitions stabilizes the underlying characteristics of the generated data. As a result, we have implemented this approach in the training process of our neural Bayes estimator. Additionally, based on the findings of Srivastava et al. (2014), randomly deactivating certain neurons during training can help prevent overfitting. Therefore, we have also introduced a dropout rate D as a hyper-parameter; refer to Algorithm 1 for a detailed description of the training algorithm.

As described in Algorithm 1, the training stage involves numerous sampling repetitions of $\boldsymbol{\Theta}^k$ and subsequently $\mathbf{Z}$ to achieve a uniform and sufficient coverage of the ranges of all parameters. Consequently, the Neural Bayes Estimator may need millions of training samples and a considerable amount of time to be properly trained. However, once the estimator is trained, the inference process becomes quite efficient because it is a straightforward plug-in procedure, which is also called *amortized inference* (Sainsbury-Dale et al. 2025).

Additionally, when simulating $\boldsymbol{\Theta}$ from its prior distribution, it is important to avoid (δ_1, δ_2) being close to 0 at the same time. Indeed, if (δ_1, δ_2) are both 0, then the GSUN process reduces to a conventional Gaussian process and, therefore, the parameters (β_2, ν_2) become void and non-estimable.

From a modeling perspective, this implies that the latent process $W^+(\mathbf{s})$ no longer contributes to the joint dependence structure, and all skewness-induced effects vanish. In such cases, the neural Bayes estimator naturally reverts to producing predictions consistent with the Gaussian process that best approximates the underlying mildly skewed process through a shifted mean vector and a perturbed covariance matrix.

5 Architecture and Spatial Data Representation

In this section, we present the two main blocks of our neural Bayes estimator for the GSUN process.

5.1 Graphical Attention Networks

Graphical Neural Networks (GNNs) arise naturally as a solution in the case of spatial processes. The realization of the study area can be represented directly as an undirected graph, where the nodes are specific locations, and the values are the corresponding realizations. The edges are drawn between two nodes if they are neighbors, thus representing the spatial information. Recent advancement in graphical neural networks (Kipf and Welling 2016; Gilmer et al. 2017) provided a framework to extend deep learning towards graph-structured data whether the goal is a node-level, edge-level or graph-level task. Among all graphical neural networks, the Graphical Attention Networks (GATs) by Veličković et al. (2018) significantly outperformed the previous GNN architectures such as the Graphical Convolutional Networks (GCNs) by Kipf and Welling (2016). The main novelty of GATs is the introduction of an attention mechanism that allows the model to adaptively weight node neighbors concerning their importance; hence, more flexible and expressive representations are possible.

Consider an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of edges. Let $N_i = \{j \in \mathcal{V}|(i, j) \in \mathcal{E}\}$ denote the set of neighbors of node i. Each node $i \in \mathcal{V}$ is associated with a feature vector $\mathbf{F}_i \in \mathbb{R}^f$, where f is the dimensionality of the input feature space. The goal of GCNs is to learn the node embeddings $\mathbf{v}_i \in \mathbb{R}^{f'}$, where f' is the dimensionality of the output feature space, by iteratively aggregating feature information from neighboring nodes. A common GCN layer aggregates node information by performing a weighted sum of neighboring node features:

$$\mathbf{F}_i^{(l+1)} = \sigma \left(\sum_{j \in N_i \cup \{i\}} \frac{1}{\sqrt{d_i d_j}} \mathbf{G}^{(l)} \mathbf{F}_j^{(l)} \right),$$

where $\mathbf{G}^l$ is a trainable weight matrix at layer l, d_i and d_j denote the degrees of node i and j, and σ is a nonlinear activation function. Although this method takes advantage of information from a local neighbourhood, it does so in a very uniform way, without discriminating between different nodes' neighbours, and this could make it difficult for the model to learn complex dependencies among nodes.

GATs remove this limitation by introducing the self-attention mechanism, allowing the nodes to assign different weights to their neighbors. For any pair of nodes i and j, we first compute the unnormalized attention coefficient:

$$e_{ij} = \text{LeakyReLU} \left\{ \mathbf{a}^\top (\mathbf{G}\mathbf{F}_i \| \mathbf{G}\mathbf{F}_j) \right\},$$

where $\mathbf{G} \in \mathbb{R}^{f' \times f}$ is a learnable weight matrix shared for all nodes, $\mathbf{a} \in \mathbb{R}^{2f'}$ is the attention vector, and $\|$ denotes the vectorization of the feature vectors of nodes i and j. Herein, $\text{LeakyReLU}(x) = \max(0.2x, x)$ introduces the non-linearity. The attention coefficient e_{ij} reflects the importance of node j's features to node i. The corresponding attention coefficients are then normalized across all neighbors of node i using the softmax function:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k \in N_i} \exp(e_{ik})}. \tag{4}$$

Considering these normalized attention coefficients, the updated representation for node i is computed as a weighted sum of the feature vectors of its neighbors:

$$\mathbf{F}_i^{(l+1)} = \sigma \left(\sum_{j \in N_i \cup \{i\}} \alpha_{ij} \mathbf{G}\mathbf{F}_j^{(l)} \right),$$

where $\sigma(\cdot)$ denotes a non-linear activation function. The attention mechanism allows the model to learn which neighbors are more relevant for the task at hand and provides more flexibility compared to GCNs.

In order to enhance the learning capacity of the model, GAT incorporates multi-head attention: it deploys K independent attention mechanisms to calculate multiple series of attention coefficients; all resulting node representations obtained from each head are then combined by concatenation (intermediate layers) or averaging (when the layer is the last). Formally, multi-head mechanisms were defined as:

$$\mathbf{F}_i^{(l+1)} = \|_{k=1}^K \sigma \left(\sum_{j \in N_i \cup i} \alpha_{ij}^{(k)} \mathbf{G}^{(k)} \mathbf{F}_j^{(l)} \right),$$

where $\|_{k=1}^K$ denotes the vectorization operation and $\alpha_{ij}^{(k)}$ are the attention coefficients from the k-th attention head. This approach improves the model's expressiveness because the model can grasp different aspects of node relationships through multiple attention heads.

5.2 Encoder Transformer

The state-of-the-art Encoder architecture (Vaswani 2017) has founded many approaches for modeling dependence structures, of which the main novelty is in using self-attention mechanisms to efficiently model long-range dependencies without any recurrence nor convolution.

The architecture of the Encoder consists of a stack of L identical layers, each having two major sub-components: a multi-head self-attention mechanism and a feed-forward neural network (FFN).

The self-attention mechanism enables the model to learn the latent dependence structure of the data by first projecting

the input data into three distinct spaces: the query ($\mathbf{Q}$), key ($\mathbf{K}$), and value ($\mathbf{V}$), where

$$\mathbf{Q} = \mathbf{X}\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{X}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{X}\mathbf{W}_V,$$

with $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d \times d_k}$ as learnable weight matrices. Then, the mechanism computes the scaled outer product between the queries and keys, which is defined as the attention scores:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d_k}}\right)\mathbf{V}.$$

Here, softmax($\cdot$) is identical to the softmax function mentioned in equation (4) but is applied row-wise to $\mathbf{Q}\mathbf{K}^\top/\sqrt{d_k}$. The attention scores are divided by $\sqrt{d_k}$ to ensure identical scales of the outer products and therefore an even distribution of the attention scores when a softmax function is applied. In addition, such an operation also stabilizes the gradients during training.

The Encoder utilizes the multi-attention mechanism as a method for enlarging the model's capacity for capturing relations of various types. Instead of one Attention block, it has h parallel attention heads, each performing attention with an independent query, key, and value matrices. The outputs of attention serve as vectors that are further concatenated and linearly transformed:

$$\mathbf{h} = \text{MultiHead}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = [\text{head}_1, \dots, \text{head}_h]\mathbf{W}_O,$$

where each attention head is computed as:

$$\text{head}_i = \text{Attention}(\mathbf{Q}_i, \mathbf{K}_i, \mathbf{V}_i),$$

and $\mathbf{W}_O \in \mathbb{R}^{hd_k \times d}$ is a learnable output projection matrix.

Following the multi-head self-attention, each layer of the Encoder applies a FFN to the learned embeddings. In particular, this FFN consists of two linear transformations with a ReLU activation in between:

$$\text{FFN}(\mathbf{h}) = \text{ReLU}(\mathbf{h}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2,$$

where $\mathbf{W}_1 \in \mathbb{R}^{d \times d_{ff}}$, $\mathbf{W}_2 \in \mathbb{R}^{d_{ff} \times d}$, and d_{ff} is the dimensionality of the hidden layer. The nonlinear transformation added by the FFN component further increases the capacity of the model and projects the embeddings to a higher dimension before lowering the dimension again to d.

To stabilize training and improve convergence, both the multi-head attention and FFN layers output are passed through a layer normalization operation following the work of Ba (2016), and residual connections are added to promote gradient flow. Formally, the output of the l-th encoder layer

is computed as:

$$\mathbf{h}_i^{(l)} = \text{LayerNorm}\left(\mathbf{h}_i^{(l-1)} + \text{MultiHead}(\mathbf{h}_i^{(l-1)})\right),$$
$$\mathbf{h}_i^{(l+1)} = \text{LayerNorm}\left(\mathbf{h}_i^{(l)} + \text{FFN}(\mathbf{h}_i^{(l)})\right).$$

These residual connections improve the model's ability to propagate gradients during back-propagation, making deeper models easier to train.

5.3 Graphical Representations of the Spatial Data and the Neural Bayes Estimator Network

In this section, we visually illustrate the graphical representation of the spatial data in detail and demonstrate the architecture of the neural Bayes estimator for the GSUN spatial process.

5.3.1 Spatial Data

Figure 1 depicts 2-D spatial data denoted with an undirected graph $\mathcal{G} = (\mathcal{V}, \xi)$. This graphical representation is obtained by plotting the locations of the spatial data on the domain of interest and treating each location as a node; we connect two nodes (locations) if their distance is within a pre-specified radius, R. Here, each node embedding $\mathbf{E}_i = (z_i, x_i, y_i)^\top$ with $i = 1, \dots, n$, where z_i is a realization of the spatial process at location $(x_i, y_i) \in [0, 1]^2$. When training the neural Bayes estimator, we set $R = 0.34$ so that the GATs can reach a sufficient coverage of the study region within 3 steps of aggregation (3 layers). Lastly, the edges ξ represent connectivity only because spatial information has already been contained in the node-level embeddings and formatted into the input of the neural Bayes estimator, see Figure 2.

5.3.2 Network Architecture

In this section, we visualize the architecture of the neural Bayes estimator in Figure 2. First, we format the input as two parts, $\mathcal{G}$ and $\mathbf{D}$, where $\mathcal{G}$ is the graphical representation (defined in Section 5.3.1) of the spatial data and $\mathbf{D} \in \mathbb{R}^{n \times n}$ is the distance matrix for their corresponding locations. There are two paths of forward propogation in this network. First, $\mathcal{G}$ is passed through 3 layers of GATs, where each layer contains 8 attention heads (n-head = 8) and outputs the node-level embeddings as 32, 256, and 512 dimensional vectors, respectively. Here, we set the output dimensions to powers of 2 for maximum. General Processing Unit (GPU)) capacity at the training stage.

Then, the node-level embeddings are stacked in a column-wise manner in the "VStack" block. Here, the dimensionally transformed distance matrix $\mathbf{D}$ is added to the stacked embeddings. The network, then, passes the aggregated data to a
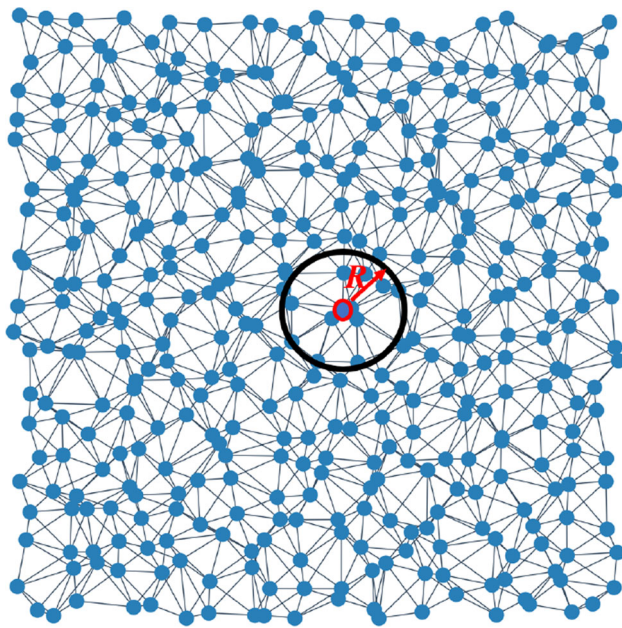
Fig. 1 Graphical representation of the spatial data with undirected edges. The circle in black denotes a pre-specified radius R to define the neighboring nodes. The red arrow denotes the radius, R

Fig. 2 GAT- and Encoder-based neural Bayes estimator for the GSUN spatial process. n-heads denotes the number of attention heads for the multi-head attention mechanism. dim demonstrates the output dimension of each GAT layer. n-layers represents the number of encoder blocks. d is the dimension of the input embeddings of the encoder block. The numbers in the parenthesis for each FFN layer illustrate its output dimension

FFN layer to be prepared as the input of the encoder block. As mentioned in Section 5.2, the encoder block, consisting of 6 encoders with 8 attention heads for each and 512 for the dimension of the input embeddings d, will further capture and extract dependencies within the data. Moreover, the output is then collapsed into a 1-D vector and passed through a FFN layer for the last prediction layer. Because there are N replicates of $(\mathcal{G}, \mathbf{D})$, the network will process each replicate in parallel, a concept that has been defined as *deepsets* in Zaheer et al. (2017). Hence, before going through the final prediction layer, the data are averaged across the batch dimension. Finally, the prediction layer maps the data to the parameter estimates $\hat{\boldsymbol{\Theta}}$. We set d and the output sizes of all FFN layers as powers of 2 for maximum GPU capacity and they are tuned according to the convergence efficiency during training. The hyper-parameter tuning of the network architecture is demonstrated in Figure 3. Figure 3a shows that having an encoder transformer block indeed helps the Bayes estimator to converge, which can be attributed to the enhanced modeling capacity and more profound understandings of the inherent dependence structure. Otherwise, the estimator experiences relatively large volatility even with 5×10^7 simulated training data. In detail, Figure 3b has shown that having more layers of a particular architecture does not always lead to better efficiency. Redundant layers can cause the neural Bayes estimator to converge at a slower rate (more simulations needed to be properly trained).
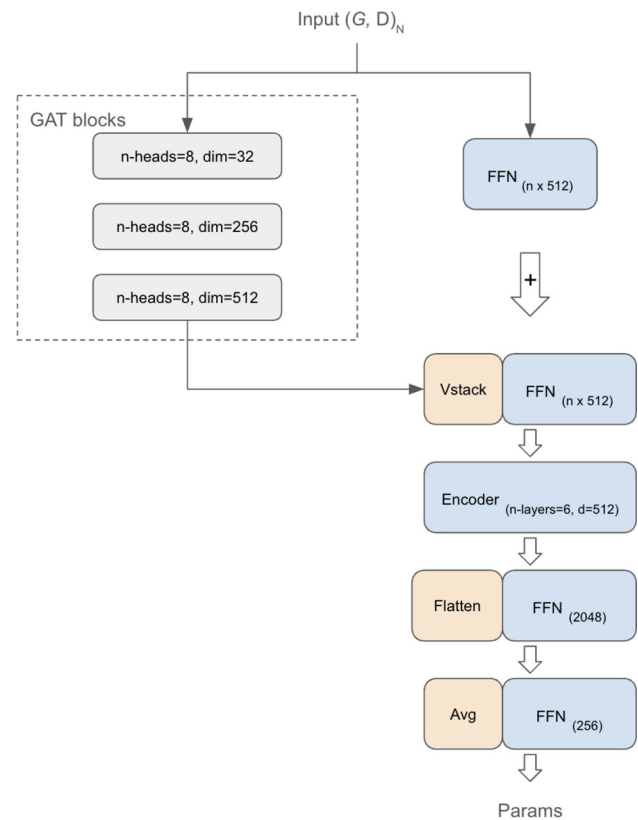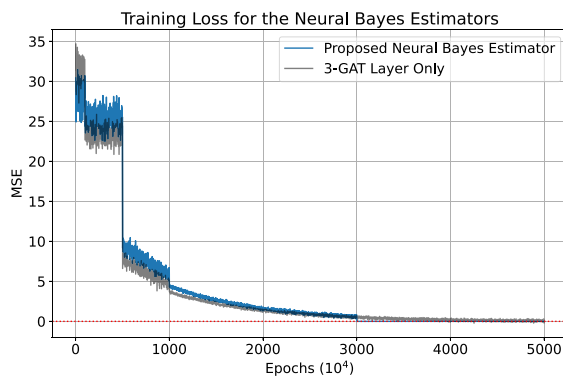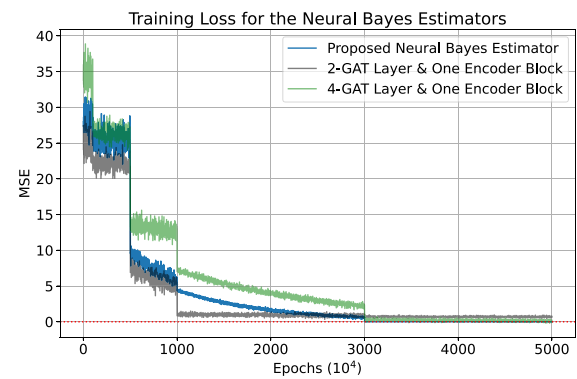
6 Simulation Study

In this section, we illustrate and analyze the performance of our proposed neural Bayes estimator for the GSUN spatial process. First, we compare the inference performance of our network with a CNN-based network architecture, which has been used for amortized inference in Goodfellow et al. (2016) and Sainsbury-Dale et al. (2024) for Gaussian processes. Second, we show that the GSUN process has its own distinct features through the probability integral transform (PIT) when compared to Gaussian and Tukey g-and-h (Xu and Genton 2017) processes. Lastly, we demonstrate the uncertainty quantification of our neural Bayes estimator and verify its validity.

6.1 Analysis of inference results

We compare the inference results of our neural Bayes estimator with the inference results obtained from a CNN-based estimator, see Figure S1 in Section S.2 of the Supplementary

(a) Training loss for the proposed neural Bayes estimator with and without an encoder block.

Fig. 3 Hyper-parameter tuning for the proposed neural Bayes estimator. The red line denotes the threshold 10^{-5}. The learning rates start as 10^{-3} and are manually adjusted (multiplied by 0.1) at

(b) Training loss curves for the proposed neural Bayes estimators with various GAT layers.

$(100, 500, 1000, 3000) \times 10^4$ epochs. The radius R is adapted to the differing numbers of GAT layers such that the aggregations can sufficiently cover the study region within the specified number of layers

Materials. In addition, Section S.2 also includes a detailed description and explanation of the network structure and associated ammortized inference. Both Bayes estimators are sufficiently and identically trained such that their empirical Bayes risks fall below 10^{-5} for a sufficiently large number of consecutive simulations, see Figure S2 in the Supplementary Materials.

To train the neural Bayes estimator, we simulate $\sigma^2 \sim \mathcal{U}(0.3, 3)$, $\beta_1 \sim \mathcal{U}(0.01, 1)$, $\nu_1 \sim \mathcal{U}(0.3, 2)$, $\beta_2 \sim \mathcal{U}(0.01, 1)$, $\nu_2 \sim \mathcal{U}(0.3, 2)$, $\delta_1 \sim \mathcal{U}(-3, 3)$, and $\delta_2 \sim \mathcal{U}(-3, 3)$, where $\mathcal{U}(l, u)$ denotes a uniform distribution on the interval (l, u). We choose these intervals to maintain a reasonable coverage of the ranges of the true values while avoiding exploding the number of samples needed from simulation to train the network.

When $\delta_1 = 0$ and $\delta_2 = 0$, the GSUN spatial model reduces to the conventional Gaussian process. As a result, the range (β_2) and smoothness (ν_2) parameters for the latent truncated process become non-estimable. Therefore, we remove them when calculating $L(\boldsymbol{\Theta}^n, \hat{\boldsymbol{\Theta}}_{(\mathbf{z}^n, \boldsymbol{\beta})})$, which is chosen as the Mean Square Error (MSE) loss. During the training process, we relax the condition for such situations to avoid weakly estimable cases, low-quality data that can potentially pollute the neural network, by removing β_2 and ν_2 from the loss when $(\delta_1, \delta_2) \in [-0.1, 0.1]^2$.

To obtain the inference results of the trained neural Bayes estimator, we simulate $N = 1$ replicate of $\mathbf{Z} \sim \mathcal{SUN}_{n,n}(\mathbf{0}, \boldsymbol{\Sigma}(\boldsymbol{\theta}_1), \mathbf{H}, \mathbf{0}, \mathbf{C}(\boldsymbol{\theta}_2))$ with $n = 400, 900, 1600$ and $\boldsymbol{\Theta} = (1, 0.15, 1, 0.1, 0.5, 0.55, -0.3)^\top$ from randomly generated locations. During inference, one can easily restructure the data into multiple sets of $n = 100$ samples and format these datasets as replicates for input. In doing this, the estimator is not limited to a fixed sample size $n = 100$. It is

important to note that here we set the sample size $n = 100$ only for the purpose of training the estimator more efficiently.

In Figure 4, we can see that the empirical posterior distribution $\Pi(\hat{\boldsymbol{\Theta}}|\mathbf{Z}, \boldsymbol{\beta})$ for both Bayes estimators is centered around specific values close to the true values and becomes more concentrated as the sample size increases.

In addition, the neural Bayes estimator proposed in this work outperforms the conventional CNN-based structure in terms of accuracy and stability. Figure 4 illustrates that the empirical distributions of $\hat{\boldsymbol{\Theta}}$ calculated using the proposed estimator have smaller IQRs across all parameters and the corresponding means are closer to the true values compared to those computed with the CNN-based estimator for most cases. Furthermore, $\sigma^2/\beta_1^{2\nu_1}$ and $1/\beta_2^{2\nu_2}$, computed using the proposed estimator, better comply with the theory of consistent estimation for in-domain asymptotics proposed in Zhang (2004). Such advantages can be attributed to the more expressive graphical representation of spatial data and the well-crafted architectures that allow for a more profound understanding of the dependence structure and a stronger modeling capacity.

6.2 Uncertainty Quantification

In this section, we explore the uncertainty quantification method for our proposed neural Bayes estimator. In Sainsbury-Dale et al. (2025), the authors have mentioned that one way to quantify the uncertainty of the neural Bayes estimator is to train another neural Bayes estimator that can learn various quantiles of the estimates based on the given sample realizations. Nonetheless, such method requires training numerous independent networks because it is difficult for a particular network to learn to output all quantiles
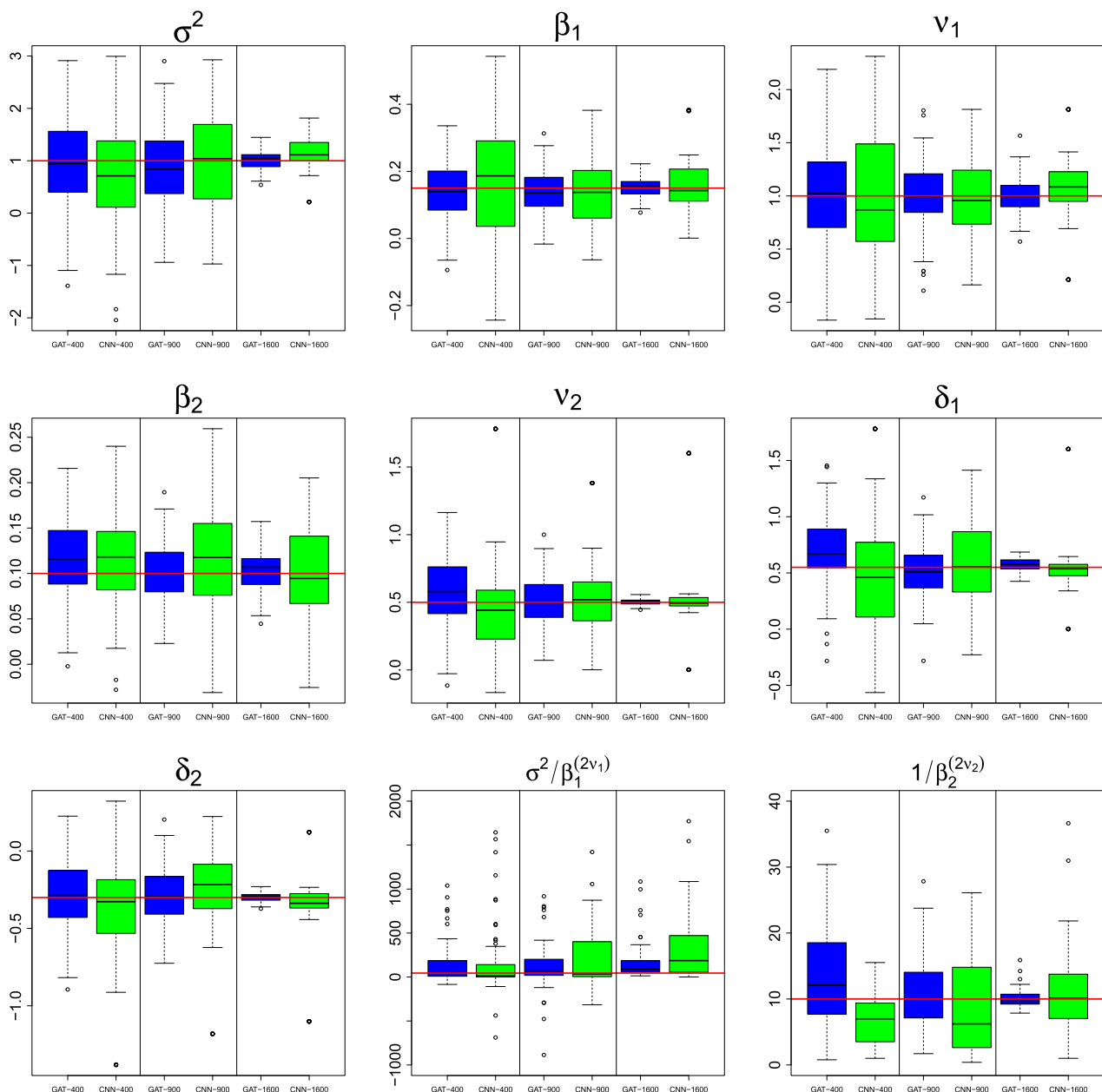
Fig. 4 The boxplots of the parameter estimates obtained from the proposed neural Bayes estimator (marked in Blue and denoted as GAT-*n*) and the CNN-based Bayes estimator (marked in Green and denoted as CNN-*n*) for $N = 500$ replicates of $n = 400, 900, 1600$ realizations of the GSUN process in $[0, 1]^2$ simulated from $\Theta = (1, 0.15, 1, 0.1, 0.5, 0.55, -0.3)^\top$. The red lines denote the true values

(continuous) at once. Therefore, in this work, we present a simulation-based approach, which can accurately and efficiently approximate the quantiles and originate from a similar idea to the bootstrap approach proposed in Sainsbury-Dale et al. (2024). A detailed description is illustrated in Algorithm 2.

To validate our proposed algorithm, we generate one replicate of $\mathbf{Z}_n$ with $\Theta = (2, 0.03, 0.5, 0.3, 1, -0.7, 0.5)^\top$ with $n = 400, 900, 1600$ and then apply Algorithm 2. To demon-

strate the accuracy of our uncertainty quantification method, we plot the empirical distributions of $\hat{\Theta}^k_{(\tilde{\mathbf{z}}_n, \boldsymbol{\beta})}$ for $k = 500$ with a bootstrap sample size $j = 1000$, together with the empirical distributions of $\hat{\Theta}^N_{(\mathbf{z}_n, \boldsymbol{\beta})}$, where $N = 500$, for coverage analysis.

Figure 5 demonstrates that the empirical distributions of $\hat{\Theta}^k_{(\tilde{\mathbf{z}}_n, \boldsymbol{\beta})}$ becomes increasingly similar to the empirical distributions of $\hat{\Theta}^N_{(\mathbf{z}_n, \boldsymbol{\beta})}$ as sample size n increases. In particular,
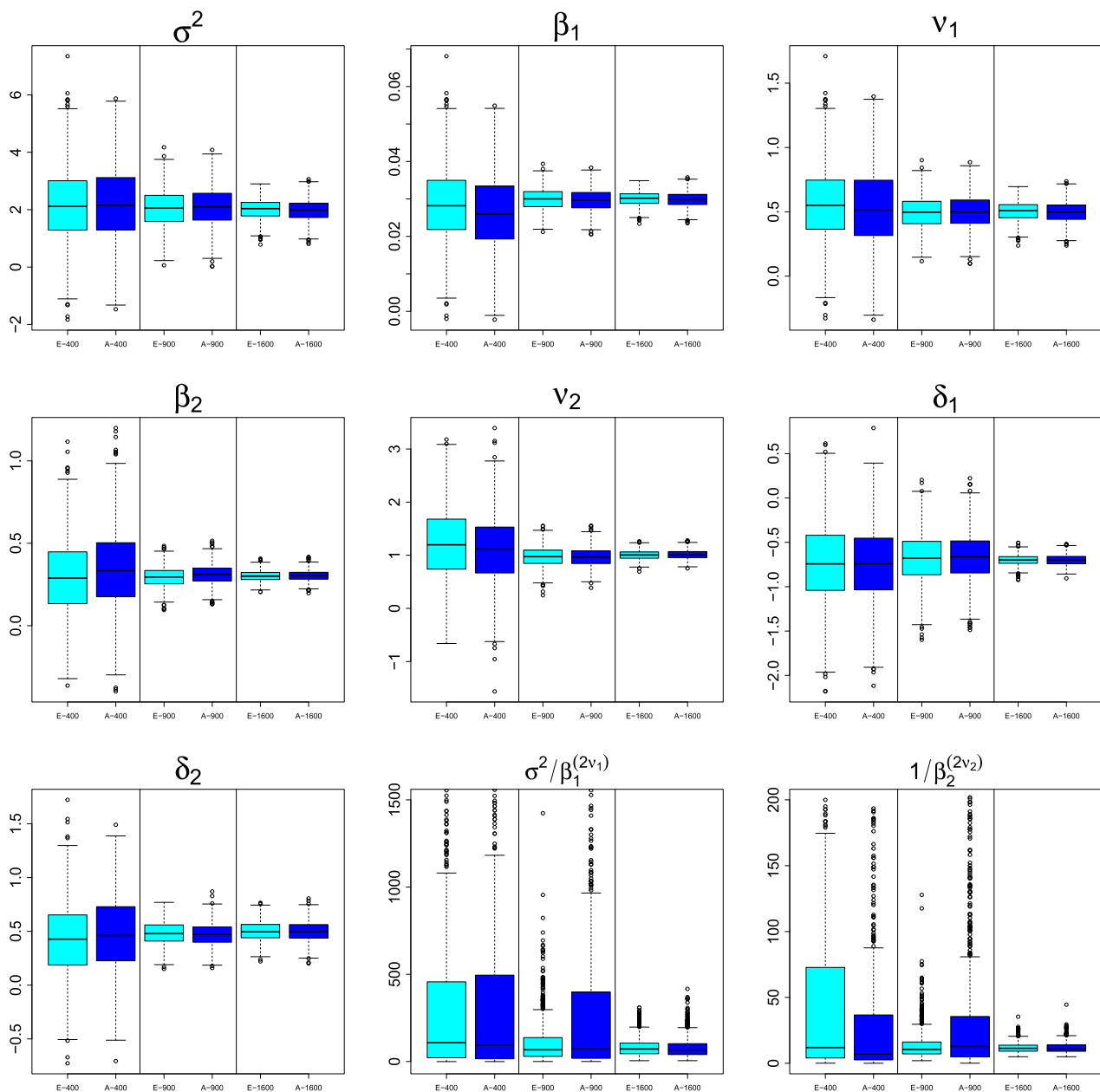
Fig. 5 The empirical densities of $\hat{\Theta}^k_{(\mathbf{z}_n,\boldsymbol{\beta})}$ (marked in Cyan and denoted as E-*n*) and $\hat{\Theta}^k_{(\tilde{\mathbf{z}}_n,\boldsymbol{\beta})}$ (marked in Blue and denoted as A-*n*) obtained from the proposed neural Bayes estimator with $N = 500$ replicates of $n = 400, 900, 1600$ realizations of the GSUN process in $[0, 1]^2$ simulated from $\boldsymbol{\Theta} = (2, 0.03, 0.5, 0.3, 1, -0.7, 0.5)^\top$ and Algorithm 2, respectively

when $n = 1,600$, the two empirical distributions almost precisely overlap for all parameters except for $(\sigma^2, \beta_1, \nu_1)$. For $(\sigma^2, \beta_1, \nu_1)$ at $n = 1,600$, there are slightly visible deviations between the two empirical distributions, but they still overlap over a large bulk. Hence, the empirical quantiles of $\hat{\Theta}^k_{(\tilde{\mathbf{z}}_n,\boldsymbol{\beta})}$ can serve as an accurate estimate for the quantiles of

$\hat{\Theta}^N_{(\mathbf{z}_n,\boldsymbol{\beta})}$. In addition, in our simulation study, j does not play a significant role when $j > 500$. However, if $j < 500$, the empirical distributions of $\hat{\Theta}^k_{(\tilde{\mathbf{z}}_n,\boldsymbol{\beta})}$ can considerably deviate from the empirical distributions of $\hat{\Theta}^N_{(\mathbf{z}_n,\boldsymbol{\beta})}$ due to potential biases. Hence, it would be safer to set j to be a large number.

6.3 Comparative Study with Gaussian and Tukey *g*-and-*h* Processes

In this section, we conduct a comparative study on the GSUN random field in contrast to the Gaussian and Tukey *g*-and-*h* random fields (Xu and Genton 2017) regarding the probability integral transformations (PIT) to demonstrate its practicality.

Gaussian random fields are commonly used to model spatial data and dependence structures (Haran 2011). They are based on a trend structure and a valid covariance function, such as the Matérn covariance family, where various parameterizations are discussed in Wang et al. (2023a). Their practicality has been repeatedly verified in real-life applications. In this regard, it is consequential to note that the Gaussianity assumption has intrinsic limitations in terms of symmetry and tail weight, which may not hold for many realistic datasets that exhibit skewness and heavy-tail properties, such as precipitation (Mondal et al. 2023).

To address these concerns, Xu and Genton (2017) proposed Tukey *g*-and-*h* random fields, defined as follows:

$$Y(\mathbf{s}) = \xi + \mathbf{X}(\mathbf{s})^\top \boldsymbol{\beta} + \omega T(\mathbf{s}),$$

where $\xi \in \mathbb{R}$ is the shift parameter, $\omega > 0$ is a scale parameter, and $\mathbf{X}(\mathbf{s})$, $\boldsymbol{\beta} \in \mathbb{R}^p$ are the covariate vector observed at location $\mathbf{s}$ and its regression coefficient. Here, $T(\mathbf{s}) = \tau_{g,h}\{Z(\mathbf{s})\}$ and $\tau_{g,h}(z) = g^{-1}(\exp(gz)-1)\exp(hz^2/2)$ is the Tukey's *g*-and-*h* transformation introduced in Tukey (1977), with $Z(\mathbf{s})$ as a Gaussian process. In this case, g dictates the amount and direction of skewness (to the right if $g > 0$ and to the left if $g < 0$) and h governs the tail behavior (where a larger h leads to heavier tail weights). Lastly, if a random vector $\mathbf{Z} \sim \mathcal{N}_n(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n)$, $\mathbf{T} = \tau_{g,h}(\mathbf{Z}) \sim \mathcal{GH}_n(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n, g, h)$, where $\mathcal{GH}_n(\boldsymbol{\mu}_n, \boldsymbol{\Sigma}_n, g, h)$ denotes a multivariate Tukey *g*-and-*h* distribution with shift $\boldsymbol{\mu}_n$, dispersion matrix $\boldsymbol{\Sigma}_n$, and g, h for skewness and tail-heaviness.

To apply the Probability Integral Transform (PIT), we need to calculate the cumulative distribution functions (cdfs) for each case. The cdf for the Gaussian distribution is well

---

Algorithm 2 Uncertainty Quantification of Neural Bayes Estimator

---

Require: Trained neural Bayes estimator $\hat{\boldsymbol{\beta}}$, graphical spatial data $(\mathcal{G}, \mathbf{D})$, number of bootstrap samples j, number of replicates k
Procedure:
   (1) Take j bootstrap samples on $(\mathcal{G}, \mathbf{D})$
   (2) Input $(\mathcal{G}, \mathbf{D})_i, \forall i = 1, \ldots, j$ into $\hat{\boldsymbol{\beta}}$ to obtain $\hat{\boldsymbol{\Theta}}^i$
   (3) Compute the average of $\hat{\boldsymbol{\Theta}}^i$ for $i = 1, \ldots, j$ and denote it as $\bar{\boldsymbol{\Theta}}$
   (4) Simulate k replicates of $\tilde{\mathbf{Z}} \sim f(\tilde{\mathbf{Z}}|\bar{\boldsymbol{\Theta}})$ and denote it as $\tilde{\mathbf{z}}^k$
   (5) Input $\tilde{\mathbf{z}}^k$ independently into $\boldsymbol{\beta}$ to obtain $\hat{\boldsymbol{\Theta}}^k_{(\tilde{\mathbf{z}}, \boldsymbol{\beta})}$
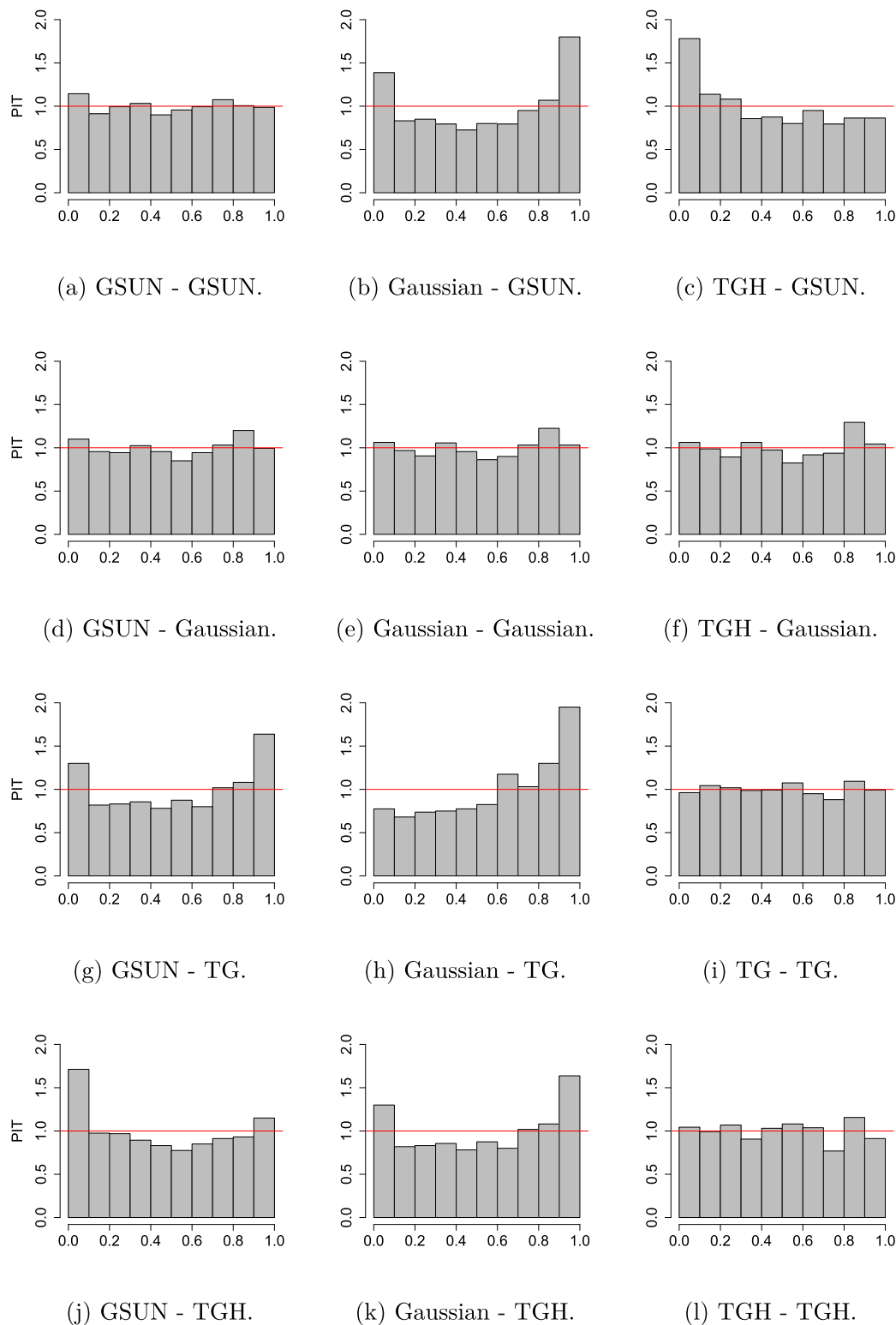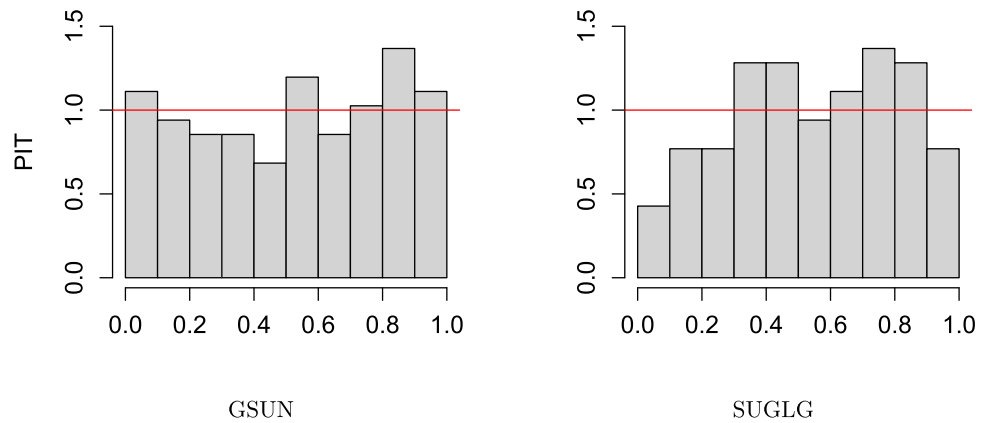   (6) Compute the empirical quantiles using $\hat{\boldsymbol{\Theta}}^k_{(\tilde{\mathbf{z}}, \boldsymbol{\beta})}$

---

known and used regularly. The cdf for the GSUN random field can be easily derived using Proposition 4 in the Supplementary Materials. For the TGH random field, we can apply the inverse Tukey's *g*-and-*h* transformation $\tau_{g,h}^{-1}\{Y(\mathbf{s})\}$ to obtain the underlying Gaussian field $Z(\mathbf{s})$ and then use the Gaussian cdf for PIT.

Moreover, we can use the Maximum Likelihood Estimation (MLE)-based inference as implemented in Abdulah et al. (2018) and Mondal et al. (2023) for parameter estimation of the Gaussian and TGH spatial models.

Figure 6 demonstrates the PIT results under three different spatial models. In particular, Figure 6a shows that the PIT is quite uniformly distributed compared to the results shown in Figures 6b and 6c, which contain obvious concentrations in either the center or the tail. Hence, the GSUN spatial process has its own distinct asymmetry and tail-weight differing from the Gaussian and TGH fields.

The same conclusion can also be drawn in Figures 6g and 6j, where we can see a striking difference between TGH and GSUN. Figures 6d and 6f demonstrate that both GSUN and TGH involve the Gaussian process as a special case. In addition, Figures 6c and 6g indicate that GSUN and TG typically have a different mean and tail weight; TG has a heavier tail weight even if there is only skewness ($h = 0$) in the model. Figure 6j indicates that when h is rather large, TGH has much heavier tails than GSUN, further highlighting the difference between TGH and GSUN.

7 Application to Pb-contaminated Soil Data

In this section, we apply the GSUN to the same real data as in Zareifard and Khaledi (2013), which contains 117 observations of Pb-contaminated areas in soils of a region of north Iran, to compare GSUN with SUGLG. In detail, we fit the GSUN and the SUGLG model proposed in Zareifard and Khaledi (2013) to obtain the parameter estimates and then plot the PIT results with the fitted models.

Figure 7 shows that the PIT with the GSUN model is closer to a uniform distribution (indicated by the red horizontal line) compared to the PIT with SUGLG. The sum of squared distances between the uniform histogram and the GSUN and SUGLG histograms are 0.366 and 0.877, respectively. Such results are anticipated because the GSUN model allows for more flexibility in the latent process that involves its own range and smoothness parameters, while the SUGLG model assumes the same covariance structure for the observed and latent processes. In addition, GSUN uses one additional parameter to control skewness compared to SUGLG and the skewness matrix **H** in GSUN is an aggregation of the principal components of the covariance matrices of the observed and latent processes (the skewness matrix is set as a scalar multiple of the correlation matrix of the

(a) GSUN - GSUN.   (b) Gaussian - GSUN.   (c) TGH - GSUN.

(d) GSUN - Gaussian.   (e) Gaussian - Gaussian.   (f) TGH - Gaussian.

(g) GSUN - TG.   (h) Gaussian - TG.   (i) TG - TG.

(j) GSUN - TGH.   (k) Gaussian - TGH.   (l) TGH - TGH.

Fig. 6 The PIT results for a sample realization of the GSUN, Gaussian, Tukey g (TG), and Tukey g-and-h (TGH) processes simulated from $\Theta = (2, 0.03, 0.5, 0.01, 1, 1, -3)^\top$ for the GSUN, $\theta_g = (\sigma^2, \beta, \nu)^\top = (2, 0.03, 0.5)^\top$ for the Gaussian, $\theta_{tg} = (\sigma^2, \beta, \nu, g)^\top = (2, 0.03, 0.5, 0.3)^\top$, and $\theta_{tgh} = (\sigma^2, \beta, \nu, g, h)^\top = (2, 0.03, 0.5, 0.3, 0.4)^\top$ for the TG and TGH at $n = 1600$ randomly generated locations on $[0, 1]^2$ using the GSUN, Gaussian, TG, and TGH models. The caption, Model - Model, indicates that we apply the cdf of the former model to the realization generated using the latter

Fig. 7 The PIT results for the 117 observations of Pb-contaminated areas in soils of a region of north Iran using GSUN and SUGLG spatial models. The red horizontal line indicates a uniform density

observed process in SUGLG), carrying more statistical intuition and interpretability, and therefore allowing for a better fit of the data.

8 Conclusion

In this work, we used a more convenient parameterization of the SUN distribution. Here, $\mathbf{H}$ represents the direction of skewness, providing a straightforward representation. This parameterization also helps to avoid the numerical instability issue from the original parametrization. Additionally, we proposed a GSUN spatial model derived from a concise reparameterization. This model ensures vanishing correlations at large distances and offers a more statistically interpretable skewness matrix. The GSUN process is more general compared with the conventional Gaussian process model. It includes the Gaussian process as a special case and models the range and smoothness of the latent process that governs the skewness, in contrast to keeping them fixed as in Zareifard and Khaledi (2013).

Due to the complexity of the parameter inference of the SUN distribution, we adopted a neural Bayes estimator to obtain point summaries of the parameters. This was achieved using a GAT- and Encoder-structured neural network while minimizing the Bayes risk. The neural Bayes estimator can be time-consuming in the training process and requires millions of simulations. However, once trained sufficiently, the inference stage is quite efficient. To this end, the proposed neural Bayes estimator in this work outperforms the conventional CNN-based architectures in terms of accuracy and stability, indicating the more enhanced modeling capacity of GAT and Encoder together with the graphical representation of spatial data.

Moreover, we compared the GSUN model with Gaussian and Tukey *g*-and-*h* models, two popularly used spatial models, to demonstrate its uniqueness through PIT plots. We applied the GSUN spatial process to 117 observations of

Pb-contaminated areas in soils of a region of north Iran and showcased its better fit over the SUGLG process.

Although the proposed GSUN framework and neural Bayes estimator demonstrated strong flexibility and predictive performance, several natural extensions and limitations merit further investigation.

First, the current formulation was developed for spatially continuous domains with a single latent skewness field. Extending the GSUN process to spatio-temporal or multivariate spatial settings would enable the joint modeling of space-time dependence and cross-covariance structures, potentially revealing additional advantages of the GSUN representation.

Second, although the neural Bayes estimator provides substantial computational efficiency once trained, it requires a large number of simulated realizations during the training phase. Future work may consider variance reduction strategies, transfer learning, or amortized inference approaches to alleviate this computational burden and improve scalability to high-dimensional spatial datasets.

Finally, the present study assumed Matérn covariance functions for both observed and latent processes. Incorporating alternative covariance families or nonstationary kernels could further improve modeling flexibility and allow the GSUN framework to capture more complex spatial heterogeneity, particularly when the assumption of isotropy is not realistic.

References

Abdulah, S., Ltaief, H., Sun, Y., Genton, M.G., Keyes, D.E.: ExaGeoStat: a high performance unified software for geostatistics on manycore systems. IEEE Trans. Parallel Distrib. Syst. **29**(12), 2771–2784 (2018)

Allard, D., Naveau, P.: A new spatial skew-normal random field model. Commun. Stat. Theory and Methods **36**(9), 1821–1834 (2007)

Alodat, M., Al-Rawwash, M.: The extended skew gaussian process for regression. Metron-Int. J. Stat. **72**(3), 317–330 (2014)

Alodat, M., Shakhatreh, M.K.: Gaussian process regression with skewed errors. J. Comput. Appl. Math. **370**, 112665 (2020)

Arellano-Valle, R.B., Azzalini, A.: On the unification of families of skew-normal distributions. Scand. J. Stat. **33**(3), 561–574 (2006)

Arellano-Valle, R.B., Azzalini, A.: Some properties of the unified skew-normal distribution. Stat. Pap. **63**, 461–487 (2022). https://doi.org/10.1007/s00362-023-01412-5

Arellano-Valle, R.B., Genton, M.G.: Multivariate unified skew-elliptical distributions. Chil. J. Stat. **1**(1), 17–33 (2010)

Arellano-Valle, R.B., Genton, M.G.: Multivariate Statistics Beyond Normality. Chapman and Hall/CRC (2026)

Azzalini, A.A.: The R package sn: the skew-normal and related distributions such as the skew-*t* and the SUN (version 2.1.1). Università degli Studi di Padova, Italia. Home page: http://azzalini.stat.unipd.it/SN/ (2023)

Ba, J.L.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)

Benavoli, A., Azzimonti, D., Piga, D.: Skew gaussian processes for classification. Mach. Learn. **109**(9), 1877–1902 (2020)

Bevilacqua, M., Caamaño-Carrillo, C., Arellano-Valle, R.B., Morales-Oñate, V.: Non-gaussian geostatistical modeling using (skew) t processes. Scand. J. Stat. **48**(1), 212–245 (2021)

Bjorck, N., Gomes, C.P., Selman, B., Weinberger, K.Q.: Understanding batch normalization. Adv. Neural Inf. Proc. Syst. **31**, (2018)

Domínguez-Molina, J., González-Farías, G., Gupta, A.: The multivariate closed skew normal distribution. Technical Report 03–12 (2003)

Genton, M.G., Zhang, H.: Identifiability problems in some non-gaussian spatial random fields. Chil. J. Stat. **3**(2), 171–179 (2012)

Gholamalinezhad, H., Khosravi, H.: Pooling methods in deep neural networks, a review. arXiv preprint arXiv:2009.07485 (2020)

Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: 34th International Conference on Machine Learning (ICML), (2017)

Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT press (2016)

Gupta, A.K., Aziz, M.A., Ning, W.: On some properties of the unified skew normal distribution. J. Stat. Theory and Pract. **7**, 480–495 (2013)

Haran, M.: Gaussian random field models for spatial data. Handbook of Markov Chain Monte Carlo 449–478 (2011)

Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Netw. **2**(5), 359–366 (1989)

Kim, H.-M., Mallick, B.K.: A bayesian prediction using the skew gaussian distribution. J. Stat. Plan. Inference **120**(1–2), 85–101 (2004)

Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)

Lau, M.M., Lim, K.H.: Review of adaptive activation function in deep neural network. In: 2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES), pp. 686–690. IEEE (2018)

Lenzi, A., Bessac, J., Rudi, J., Stein, M.L.: Neural networks for parameter estimation in intractable models. Comput. Stat. Data Anal. **185**, 107762 (2023)

Mahmoudian, B.: On the existence of some skew-gaussian random field models. Stat. Probab. Lett. **137**, 331–335 (2018)

Márquez-Urbina, J.U., González-Farías, G.: A flexible special case of the CSN for spatial modeling and prediction. Spat. Stat. **47**, 100556 (2022)

Minozzo, M., Ferracuti, L.: On the existence of some skew-normal stationary processes. Chil. J. Stat. **3**(2), 157–170 (2012)

Mondal, S., Abdulah, S., Ltaief, H., Sun, Y., Genton, M.G., Keyes, D.E.: Tile low-rank approximations of non-gaussian space and space-time tukey g-and-h random field likelihoods and predictions on large-scale systems. J. Parallel and Distrib. Comput. **180**, 104715 (2023)

Nagi, J., Ducatelle, F., Di Caro, G.A., Cireşan, D., Meier, U., Giusti, A., Nagi, F., Schmidhuber, J., and Gambardella, L.M.: Max-pooling convolutional neural networks for vision-based hand gesture recognition. In: 2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA), pp. 342–347. IEEE (2011)

Richards, J., Sainsbury-Dale, M., Zammit-Mangion, A., Huser, R.: Likelihood-free neural bayes estimators for censored peaks-over-threshold models. J. Mach. Learn. Res. **25**, 1–49 (2024)

Rimstad, K., Omre, H.: Skew-gaussian random fields. Spat. Stat. **10**, 43–62 (2014)

Sainsbury-Dale, M., Richards, J., Zammit-Mangion, A., Huser, R.: Neural bayes estimators for irregular spatial data using graph neural networks. J. Comput. Graph. Stat. **34**, 1153–1168 (2025)

Sainsbury-Dale, M., Zammit-Mangion, A., Huser, R.: Likelihood-free parameter estimation with neural bayes estimators. Am. Stat. **78**(1), 1–14 (2024)

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. The J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)

Tagle, F., Castruccio, S., Crippa, P., Genton, M.G.: A non-gaussian spatio-temporal model for daily wind speeds based on a multivariate skew-t distribution. J. Time Ser. Anal. **40**(3), 312–326 (2019)

Targ, S., Almeida, D., Lyman, K.: Resnet in resnet: generalizing residual architectures. arXiv preprint arXiv:1603.08029 (2016)

Tukey, J.W., et al.: Exploratory Data Analysis, vol. 2. Springer (1977)

Vaswani, A.: Attention is all you need. Adv. Neural Inf. Proc. Syst. **30**, 1 (2017)

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: 6th International Conference on Learning Representations (ICLR), (2018)

Walchessen, J., Lenzi, A., Kuusela, M.: Neural likelihood surfaces for spatial processes with computationally intensive or intractable likelihoods. Spat. Stat. **62**, 100848 (2024)

Wang, K., Abdulah, S., Sun, Y., Genton, M.G.: Which parameterization of the Matérn covariance function? Spat. Stat. **58**, 100787 (2023)

Wang, K., Arellano-Valle, R.B., Azzalini, A., Genton, M.G.: On the non-identifiability of unified skew-normal distributions. Stat. **12**(1), e597 (2023)

Wang, K., Karling, M.J., Arellano-Valle, R.B., Genton, M.G.: Multivariate unified skew-t distributions and their properties. J. Multivar. Anal. **203**, 105322 (2024)

Wilhelm, S., Manjunath, B.: tmvtnorm: truncated multivariate normal and student t distribution. R Package Version **1**, 6 (2023)

Xu, G., Genton, M.G.: Tukey g-and-h random fields. J. Am. Stat. Assoc. **112**(519), 1236–1249 (2017)

Zaheer, M., Kottur, S., Ravanbakhsh, S., Poczos, B., Salakhutdinov, R.R., Smola, A.J.: Deep sets. Adv. Neural Inf. Proc. Syst. **30**, (2017)

Zammit-Mangion, A., Sainsbury-Dale, M., Huser, R.: Neural methods for amortised parameter inference. Annu. Rev. Stat. Its Appl. **12**, 311–335 (2025)

Zareifard, H., Khaledi, M.J.: Non-Gaussian modeling of spatial data using scale mixing of a unified skew Gaussian process. J. Multivar. Anal. **114**, 16–28 (2013)

Zhang, H.: Inconsistent estimation and asymptotically equal interpolations in model-based geostatistics. J. Am. Stat. Assoc. **99**(465), 250–261 (2004)

Zhang, H., El-Shaarawi, A.: On spatial skew-gaussian processes and applications. Environmetrics **21**(1), 33–47 (2010)

Zhou, D.-X.: Universality of deep convolutional neural networks. Appl. Comput. Harmon. Anal. **48**(2), 787–794 (2020)