Received: 26 April 2023 | Revised: 13 February 2024 | Accepted: 29 February 2024

DOI: 10.1002/env.2845

RESEARCH ARTICLE

WILEY

Fast parameter estimation of generalized extreme value distribution using neural networks

Sweta Rai[1] | **Alexis Hoffman**[2] | **Soumendra Lahiri**[3] | **Douglas W. Nychka**[1] | **Stephan R. Sain**[2] | **Soutir Bandyopadhyay**[1]

[1]Department of Applied Mathematics and Statistics, Colorado School of Mines, Golden, Colorado, USA

[2]Jupiter Intelligence, Boulder, Colorado, USA

[3]Department of Statistics and Data Science, Washington University in St. Louis, St. Louis, Missouri, USA

Correspondence
Soutir Bandyopadhyay, Department of Applied Mathematics and Statistics, Colorado School of Mines, Golden, CO, USA.
Email: sbandyopadhyay@mines.edu

Funding information
National Science Foundation, Grant/Award Numbers: CMMI-2210840, CMMI-2210811

Abstract

The heavy-tailed behavior of the generalized extreme-value distribution makes it a popular choice for modeling extreme events such as floods, droughts, heatwaves, wildfires and so forth. However, estimating the distribution's parameters using conventional maximum likelihood methods can be computationally intensive, even for moderate-sized datasets. To overcome this limitation, we propose a computationally efficient, likelihood-free estimation method utilizing a neural network. Through an extensive simulation study, we demonstrate that the proposed neural network-based method provides generalized extreme value distribution parameter estimates with comparable accuracy to the conventional maximum likelihood method but with a significant computational speedup. To account for estimation uncertainty, we utilize parametric bootstrapping, which is inherent in the trained network. Finally, we apply this method to 1000-year annual maximum temperature data from the Community Climate System Model version 3 across North America for three atmospheric concentrations: 289 ppm CO_2 (pre-industrial), 700 ppm CO_2 (future conditions), and 1400 ppm CO_2, and compare the results with those obtained using the maximum likelihood approach.

KEYWORDS

deep neural networks, extreme quantiles, generalized extreme value distribution, parameter estimation, sufficient statistics

1 | INTRODUCTION

The generalized extreme value (GEV) distribution serves as a versatile statistical model used to model extreme events like floods, temperatures, and precipitation (Engeland et al., 2004; Huang et al., 2016; Stojkovic & Simonovic, 2019; Wang et al., 2016). Efficiently estimating its parameters and quantiles is crucial for precise characterization and effective analysis of extreme behavior. Its widespread applicability makes it valuable in various disciplines for quantifying and understanding rare and extreme events. Various traditional techniques, including the method of moments (MOM), maximum likelihood (ML) estimation approach, probability weighted moments (PWM), and L-moments estimation, are conventionally employed to estimate the parameters of the GEV distribution. Comparative studies have been conducted to discern the advantages of these approaches over one another Prescott and Walden (1980), Hosking et al. (1985), Hosking (1985), and Coles and Dixon (1999).

Fitting of the GEV distribution involves determining its location, scale, and shape parameters. However, estimating the shape parameter of the GEV with the ML approach can lead to underestimation, especially with small sample sizes, as observed by Coles and Dixon (1999) and consistent with Hosking et al. (1985). To address this issue, Martins and Stedinger (2000) proposed the use of Bayesian priors within a generalized maximum likelihood (GML) analysis to restrict the range of shape parameters in the likelihood estimation. Therefore, careful consideration is required when employing the ML approach for small samples, and computational constraints should be considered while using the likelihood approach for extensive simulations. On the other hand, moments-based approaches like the MOM and PWM are less efficient compared to the ML approach and require subjective weight selection for moderate samples.

Exploring alternative approaches, such as Bayesian methods or neural estimators like deep learning algorithms, offers distinct advantages over conventional methods. The neural estimators act as computational models, utilizing neural networks (NNs) to predict or estimate outcomes based on input data. One notable advantage is their computational efficiency. Similarly, in a Bayesian setting, Yoon et al. (2010) proposed a full Bayesian GEV distribution estimation approach. Each of these approaches comes with a unique set of strengths and limitations. There has been recent development of neural estimators for intractable likelihood function, see Creel (2017), Lenzi et al. (2023), Lenzi and Rue (2023), Sainsbury-Dale, Zammit-Mangion, and Huser (2023), and Sainsbury-Dale, Richards, et al. (2023). The probability density function (PDF) of the GEV distribution integrates exponentials and power functions, resulting in a nonlinear and often analytically intractable expression. The optimization of the likelihood proves challenging, particularly hinging on the shape parameter, which elucidates the tail behavior and, in turn, defines the heavy-tail nature of the GEV distribution. Building a neural estimator to estimate parameters of the GEV distribution with similar accuracy to ML or moment-based approach is crucial not only to model the marginal extremes but also to further implementation for multivariate extreme modeling (Davison et al., 2012; Huser & Wadsworth, 2022; Padoan et al., 2010). These works show promising results in accuracy as compared to the classical approaches, along with a potential speed-up factor in the overall estimation process. However, the application of these works mostly focuses on modeling spatial extremes using max-stable processes for higher dimensions.

Motivated by recent advancements in applying deep learning algorithms to extreme events, as discussed earlier, and inspired by the successful utilization of NNs for time series and spatial data, as evidenced in papers such as Cremanns and Roos (2017), Gerber and Nychka (2021), Majumder et al. (2022), and Wikle and Zammit-Mangion (2023), our research introduces a novel estimation method. In this work, we present a new estimation method that utilizes a deep NN to fit univariate GEV distributions to extreme events. It is crucial to note that identifying the marginal GEV distributions is typically required for modeling any multivariate extreme process, and therefore, in this article, we focus on the computationally efficient modeling of univariate GEV distributions.

Our focus extends beyond the development of a neural estimator. We strategically employ the utilization of sufficient statistics in our estimation process to make inferences about unknown parameters. The motivation behind incorporating informative statistics is to achieve computational efficiency and simplification. The use of informative statistics instead of full datasets for modeling extremes has been reported in several studies, including Creel and Kristensen (2013), Creel (2017), Polson and Sokolov (2023), and Sainsbury-Dale, Zammit-Mangion, and Huser (2023). By including both extreme quantiles and the standard quartiles (Q_1, Q_2, and Q_3), our NN model is able to effectively capture the important tail behavior in extreme event modeling. The utilization of sample quantiles as inputs for the network is supported by the concept of order statistics. We generate training data through simulation. During training, we input the sample quantiles and apply activation functions to generate optimal nonlinear representations, which are then used to estimate the GEV parameters. The model outputs an estimate of the GEV parameters, which define the distribution of the extreme event. To ensure a robust model, we utilize simulated values within a reasonable parameter range for training and work with a large training size. Additionally, we utilized a validation set to monitor the model performance during training.

The use of an NN for this problem offers the following benefits:

(a) The NN architecture is well suited for inference, allowing for the efficient estimation of parameters. In particular, the network can be quickly evaluated once trained, resulting in significant speed gains of up to 170 times or even more compared to traditional methods.
(b) To address the issue of uncertainty in our parameter estimates, we adopt the bootstrapping approach, which has been widely supported by previous research (see, for example, Cooley et al. (2007), Huang et al. (2016), Gamet and Jalbert (2022), Yadav et al. (2022), Lenzi et al. (2023), and Sainsbury-Dale, Zammit-Mangion, and Huser (2023)). In particular, to generate confidence intervals, we utilized the parametric bootstrap, which is typically computationally

intensive. However, the trained NN enables efficient simulation and evaluation of bootstrap samples, resulting in the rapid generation of confidence intervals.

Finally, our model is employed to analyze the maximum temperature extremes in the Community Climate System Model version 3 (CCSM3) model runs at three distinct CO_2 concentrations throughout North America. By examining annual maximum temperature data, we demonstrate the accuracy and advantages of our approach compared to the classical ML method. The fast evaluation speed of the neural estimator facilitated efficient uncertainty quantification through parametric bootstrap sampling. Our findings indicate that we can produce hundreds of spatial confidence intervals within a matter of seconds.

The remainder of the article is structured as follows. Section 2 offers an overview of the GEV distribution and elucidates the proposed NN model. Section 3 showcases the outcomes of our simulation study. Section 4 delves into our CCSM3 runs case study, and lastly, Section 5 recapitulates our findings, examines the behavior and limitations of our model, and presents our conclusion.

2 | METHODOLOGY

This section provides an overview of the structure of the GEV distribution and outlines our model framework. It also includes information on the approximate statistics chosen as inputs for the network and the network architecture used in our model.

2.1 | Generalized extreme-value distribution

The GEV distribution, introduced by Jenkinson (1955), is defined by the parameter vector $\theta = (\mu, \sigma, \xi)$. Here, location-scale parameters $(\mu, \sigma) \in R \times (0, \infty)$ and the shape or tail-index parameter $\xi \in R$. The cumulative distribution function (CDF) of the GEV distribution is denoted as:

$$F(x) = \begin{cases} \exp\left\{-[1 + \xi(x-\mu)/\sigma]^{-1/\xi}\right\}, & \text{if } \xi \neq 0 \\ \exp\{-\exp[-(x-\mu)/\sigma]\}, & \text{if } \xi = 0. \end{cases}$$

The support of F is determined by the interval

$$S_\theta = \{x \in R : \sigma + \xi(x-\mu) > 0\}. \tag{1}$$

Therefore, the CDF F is defined only for values of x that fall within S_θ. The GEV distribution can take on three different forms depending on the sign of its shape parameter, ξ. These forms are the Gumbel distribution for light-tailed distributions ($\xi = 0$), the Fréchet distribution for heavy-tailed distributions ($\xi > 0$), and the Weibull distribution for short-tailed distributions ($\xi < 0$). The sign of ξ also determines whether the GEV distribution is upper-bounded ($\xi < 0$) or lower-bounded ($\xi > 0$).

When evaluating the risk associated with extreme events in extreme value analysis (EVA), return levels are a crucial component. These levels estimate the expected values of extreme quantiles that may occur within a specific time frame, or return period, represented by T. The GEV distribution typically models block maxima data, often with a standard block size of a year. This choice simplifies the definition of return periods and makes the definition of quantiles more natural when dealing with annual max data. However, adjusting the block size introduces complexity in defining the return period, denoted as T. The equation for computing return levels is given by $z_T = F^{-1}(1 - T^{-1})$, where F^{-1} refers to the inverse CDF of the GEV distribution, and T denotes the return period. The use of the $(1 - T^{-1})$ quantile as the threshold is important because it represents the average frequency with which this threshold is exceeded over the specified return period.

Determining the large-sample asymptotics of the ML estimator for the GEV parameters is challenging due to the dependence of the support S_θ on the parameters θ. However, Smith (1985) and Bücher and Segers (2017) have shown that by restricting the lower bound of $\xi > -1/2$, demonstrate the preservation of asymptotic normality of the MLE.

2.2 | Approximate sufficient statistics

In this article, we propose an indirect inference method for estimating the unknown parameter $\theta \in \Theta \subset R^k$ of the underlying GEV distribution, using a minimal set of lower-dimensional statistics of the training sample as input to a deep NN. The approach is specifically designed to infer the heavy-tailed behavior of the GEV distribution. It achieves this by utilizing a set of extreme quantiles (from both the lower and upper ends), including the three standard quartiles Q_1, Q_2, and Q_3. The use of quantiles for inference is not new as we see the use of quantile-based deep NN in Polson and Sokolov (2023), where the authors endeavor to construct generative artificial intelligence for Bayesian computation, employing indirect mapping of data to the target parameter through the use of sufficient statistics (quantiles). In Sainsbury-Dale, Zammit-Mangion, and Huser (2023), we observe the utilization of permutation-invariant neural networks for optimal estimation. They employ a permutation-invariant mapping on the training set and input the transformed data in a DeepSet neural structure to obtain the Bayes estimator for cases of intractable likelihood. However, it is worth noting that working with the quantile function can be considered a specific instance of this permutation-invariant mapping.

This approach is also motivated by previous studies (Creel & Kristensen, 2013; Jiang et al., 2017) that have shown promising results using informative statistics in statistical modeling. Still, to the best of our knowledge, this is the first study that explores the use of extreme quantiles as input to an NN for the estimation of the GEV distribution parameters in particular. The study aims to examine the ability of a given set of quantiles to estimate the parameters of the GEV distribution using a sophisticated deep NN, with the goal of providing a more efficient and accurate method for inferring the heavy-tailed nature of the distribution. Further details about the NN framework are provided in Section 2.3.

While using summary statistics in indirect inference is a valuable approximation method, it may not consistently achieve optimal asymptotic efficiency. Addressing this concern typically involves conducting simulation studies to guide your selection. To tackle this challenge, it is essential to choose informative quantiles that effectively capture the heavy-tailed characteristics of the GEV distribution while keeping the NN's input dimensionality minimal. We explore various quantile options to identify a set that approaches optimality, closely analyzing their effects on the network's behavior and the resulting implications for efficiency, robustness, and interpretability.

2.3 | NN framework

Our estimation technique utilizes a deep NN that takes the quantile values as its inputs and returns an approximate $\theta = (\mu, \sigma, \xi) \in \Theta$ as the dependent output, where Θ represents the parameter space. Let $\mathcal{P} = \mathcal{P}(\mathbf{y}) \in \mathcal{R}^m$ be the quantile/percentile values, represented as a scalar vector, $m < n$. Here, m and n are the sizes of $\mathcal{P}(\mathbf{y})$ and $\mathbf{y}$, respectively, for sample $\mathbf{y}$. The function $\mathcal{N}$ maps $\mathcal{P}$ to θ such that $\widehat{\theta} = (\widehat{\mu}, \widehat{\sigma}, \widehat{\xi})$ is obtained as the corresponding estimate

$$\mathcal{N} : \mathcal{R}^m \to \Theta; \text{ where } \mathcal{N}(\mathcal{P}) = \widehat{\theta}.$$

We construct a feedforward neural network, where information flows unidirectionally- from the input layer through any hidden layers to the output layer. In this architecture, each layer of neurons acts as input to the next layer without forming cycles or loops. Our neural estimator, denoted as $\mathcal{N}$ has L layers, where the first layer is the input layer, the final layer is the output layer, and the remaining $L-2$ layers are the hidden layers. Let the jth layer of $\mathcal{N}$ has n_j neurons, $j = 1, 2, \ldots, L$, and is characterized by the nonlinear activation function f_j. The interaction between the successive layers is defined by the recursive equation

$$h_j = f_j(b_j + w_j h_{j-1}), \quad j = 1, 2, \ldots, L,$$

where h_j is the vector output for all neurons in the jth layer, b_j is the bias vector, w_j is the weight matrix connecting the $(j-1)$th layer to the jth layer, and h_{j-1} is the vector output of the $(j-1)$th layer. This equation captures the computation performed by each layer of the network, where the input to each layer is the output of the previous layer after being transformed by the weight matrix, bias vector, and activation function. Thus, $\mathcal{N}$ maps inputs to outputs through a sequence of nonlinear transformations performed over the subsequent layers.

It is essential to recognize the connection between NNs and generalized linear models (GLMs) (Nelder & Wedderburn, 1972), as they share a remarkable analogy in their modeling approaches. Both NNs and GLMs rely on a function to process inputs; in NNs, this function is the activation function, while in GLMs, it is the link function. The activation

function in NNs non-linearly transforms inputs, enabling the modeling of complex data relationships, while the link function in GLMs connects the linear combination of predictors to the distribution of the response variable. In practice, when training NNs, we adjust weights and biases to minimize the discrepancy between the predicted $\widehat{\theta}$ and the true θ for a given input $\mathcal{P}$. This aligns with the core objective of fitting GLMs to real-world data.

Selecting an appropriate performance loss function to measure the discrepancy between θ and $\widehat{\theta}$ is crucial in building an NN algorithm. A commonly used metric for measuring the performance of an NN algorithm is the mean squared error (MSE) loss, which is computed over a selected batch of the training sample and is expressed as follows:

$$\text{MSE}(\omega) = \frac{1}{n_B}\sum_{i=1}^{n_B}\|\theta_i - \widehat{\theta}_i(\omega)\|_2^2 = \frac{1}{n_B}\sum_{i=1}^{n_B}\left\{(\mu_i - \widehat{\mu}_i(\omega))^2 + (\sigma_i - \widehat{\sigma}_i(\omega))^2 + (\xi_i - \widehat{\xi}_i(\omega))^2\right\}, \tag{2}$$

where n_B is the batch size, ω denotes the matrix that encodes the network's weights and biases for a given iteration stage, and $\|\cdot\|_2$ is the l_2 norm. Minimizing the MSE loss during training is essential for obtaining accurate estimates of θ, thereby achieving a close prediction of the true parameter value. However, it is also important to note that alternative loss functions beyond MSE, such as mean absolute error (MAE), Huber loss, quantile loss, and others, may also be used depending on the specific requirements and characteristics of the estimation problem.

In the context of estimating parameters of the GEV distribution, it is important to ensure that the estimated parameters satisfy the support constraint of GEV denoted by Equation (1). To meet the support constraint of the GEV distribution, we use a reparameterization. Instead of directly estimating the scale parameter σ, we estimate the transformation parameter δ, along with μ and ξ. This reparameterization ensures that the estimated parameters adhere to the constraint within the GEV support, defined as $\sigma > \xi(\mu - \mathbf{y})$. Specifically, σ can be expressed as a function of μ, ξ, and δ as follows:

$$\sigma = \xi(\mu - \mathbf{y}^*) + \exp(\delta),$$

where, $\mathbf{y}^*$ represents the sample maxima if $\xi < 0$ and the sample minima if $\xi > 0$.

2.4 | Network training

To train our neural estimator $\mathcal{N}$, we generate a comprehensive training dataset by simulating values from the GEV distribution across a range of feasible parameters θ. Specifically, we generate in total of 340,000 parameter configurations for training and validation. We perform uniform sampling within the range $\mu \in (1, 50)$, $\sigma \in (0.1, 40)$, and $\xi \in (-0.4, 1)$, with the choices of μ and σ informed by our analysis of temperature, precipitation, and wind data. After establishing a parameter configuration, we simulate a GEV sample and then apply standardization. This standardization involves subtracting the sample median and scaling by the sample interquartile range (IQR), which also rescales the values of μ and σ. Notably, the restricted range of $\xi > -1/2$ aligns with the literature, making it suitable for a comparative study against classical approaches.

To investigate the impact of sample size on the estimation process and ensure the generalizability of our study, we generate GEV samples with various sample sizes within the range of 30 to 1000 across the parameter configurations. For the training and validation phases, we split the total of 340,000 parameter configurations into a training set (consisting of 300,000 configurations) and a validation set (consisting of 40,000 configurations) to monitor overfitting. We work with five different sample sizes: $n = 30, 72, 173, 416, 1000$. For each sample size, we generate GEV samples across 68,000 distinct parameter configurations within specified parameter ranges, offering a comprehensive assessment of various sample sizes for GEV sample generation, contributing to the total parameter configuration of 340,000. Utilizing a large training set is essential for optimizing the network's weights, resulting in a robust mapping from inputs to outputs. To mitigate overfitting, we implement early stopping measures during the training process, closely monitoring both the validation loss and learning rate. If no improvement in the validation loss is observed, the training process is halted.

To serve as the input to the NN, we select a suitable set of percentiles, $\mathcal{P}$. To identify an appropriate set of percentiles, we consider a range of values spanning from the 0.01th to the 99.99th percentile over the generated GEV samples given by

$$\mathcal{P} = \{\mathbf{0.01th}, \mathbf{0.1th}, \mathbf{1th}, \mathbf{10th}, 25\text{th}, 50\text{th}, 75\text{th}, \mathbf{90th}, \mathbf{99th}, \mathbf{99.9th}, \mathbf{99.99th}\}, \tag{3}$$

where we have boldfaced the extreme percentile values for emphasis.

By selecting percentiles from this wide range of values, we can capture the full spectrum of the GEV samples and better understand the heavy-tailed behavior of the distribution. However, the extreme quantiles for small samples like size 30 or 72, may be the sample maximum and minimum and may overlap. We conduct a comparative study to determine the optimal number of hidden layers in the network architecture. The results are presented in Table 1.

We employ the network architecture described in Figure 1. In training $\mathcal{N}$ illustrated in Figure 1, we investigate the use of differently sized GEV samples to enhance the fidelity of our model to real-world scenarios. The MSE loss function, as described in Section 2.3, is utilized to train and optimize our model. For optimization, we employ the root mean square propagation (RMSprop) optimizer, discussed in Hinton et al. (2012), to enhance the modeling of heavy-tailed distributions by adjusting weights for each batch of 128 samples.

RMSprop is an adaptive learning rate optimization algorithm designed to overcome limitations in traditional gradient descent methods. It adapts the learning rates, contributing to training stability and mitigating the risk of exploding gradients, especially in the presence of heavy-tailed and unstable data. The optimizer is initialized with a learning rate of 0.001, enabling efficient and effective adjustments to the model parameters during training. Table 2 provides a comprehensive overview of the architecture, including the output shape per layer and the activation functions employed.

Our neural estimator output layer employs a linear activation function, producing three scalar values corresponding to the shifted μ, δ, and ξ. For the hidden layers, we utilize the rectified linear unit (ReLU) activation function, a piecewise linear function that yields zero for negative input values and retains the input for positive values. This activation function

TABLE 1 Comparing different architectures trained using various multilayer perceptron (MLP) NNs on the same training set.

No. of hidden layers	Training parameter	Validation loss	Training loss
2-MLP	18,435	21979.52	26,857,216
3-MLP	88,707	4.59	1142.32
5-MLP	614,019	4.58	4.61
6-MLP	1,139,331	4.56	4.54

Note: In this context, "number-MLP" refers to the number of hidden layers in the architecture. After comparing different architectures, the 5-MLP network was chosen for its better performance and efficiency in minimizing the MSE.

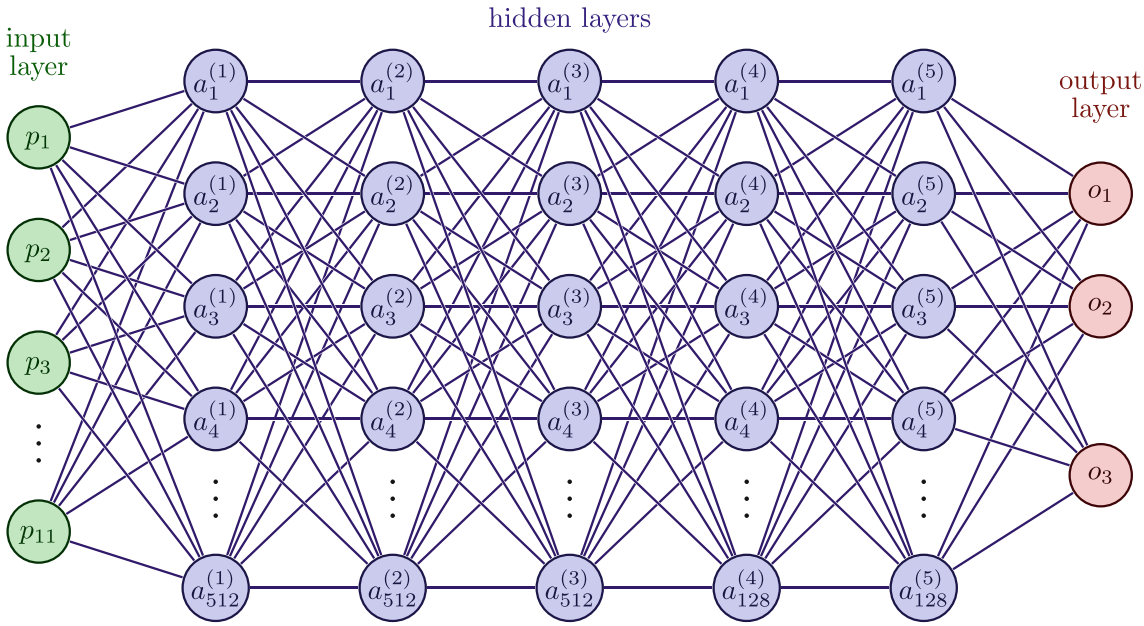

FIGURE 1 The network architecture of $\mathcal{N}$ takes $\mathcal{P}$, denoted as $p_1, p_2, \ldots, p_{11}$, as input. The notation $a_1, \ldots, a_{v_j}$ represents the nodes in the jth hidden layer (h_j), where v_j indicates the number of nodes in the layer, ranging from 1 to 5. The model produces three scalar values as output.

TABLE 2 Overview of hidden layers and output layer in our FFNN model.

Layer type	Output shape	Activation	Parameters
Dense	512	ReLU	6144
Dense	512	ReLU	262,656
Dense	512	ReLU	262,656
Dense	128	ReLU	65,664
Dense	128	ReLU	16,512
Dense	3	Linear	387

Note: The input layer has shape 11; 11 quantile values as inputs, and returning the estimated GEV parameters as output with shape 3.

is often preferred in hidden layers due to its computational efficiency, non-linearity, gradient stability, and sparsity. We incorporate early-stopping criteria based on validation MSE (as defined in Equation (2)) to prevent overfitting, halting the training process when no further improvement in loss is observed.

2.5 | Sample standardization

In this section, we will discuss the importance of standardizing the sample before training the network. The standardization process has several benefits, including improved stability, estimation, and performance. However, our main objective here is to make the network more versatile and applicable to a wider range of extreme scenarios by making it invariant to different scales and units of measurement.

We center and scale the GEV sample using the sample median and interquartile range (IQR). Let $y = (y_1, \ldots, y_n)$ be a sample of size n from the GEV(μ, σ, ξ) distribution. The standardization is expressed as

$$z = \frac{y - \tilde{y}}{\text{IQR}},$$

where $\tilde{y}$ is the sample median, IQR is the sample interquartile range, and z is the standardized GEV sample. However, rescaling the sample can alter its location (μ) and scale (σ). The adjustment of μ is given by

$$\frac{\mu - \tilde{y}}{\text{IQR}},$$

and the adjustment of σ is

$$\frac{\sigma}{\text{IQR}}.$$

To account for these changes, we train $\mathcal{N}$ with transformed percentile values and then invert the transformation to the original scale to compare with the true values.

By implementing standardization, an NN becomes more robust and better able to generalize to extreme events, such as precipitation and wind, measured in different units, making it more versatile. Standardization is typically performed using pairing methods such as mean with IQR or mean with standard deviation. In this work, we have chosen the sample median-IQR pairing, as it yields better results and stability compared to other combinations.

3 | SIMULATION STUDY

This section presents the results of simulation studies on the precision of the neural estimator $\mathcal{N}$ using a test set of N_{test} parameter configurations generated from the selected parameter range described in Section 2.4.

3.1 | Comparison with classical approaches

For comparison of the model performance, we also calculate the maximum likelihood estimates (MLEs) and L-moment (LM) estimates of the GEV parameters on the test dataset. The MLEs and LM estimates are computed using the parameter estimation methods implemented in the R packages `ismev` (Stephenson & Heffernan, 2014) and `extRemes` (Gilleland & Katz, 2016). Specifically, the `ismev` package employs the `optim` function for numerical optimization to provide the MLEs of the GEV distribution using the Nelder–Mead optimization method (see, Singer and Nelder (2009)). The accuracy of the parameter estimates from our neural estimator $\mathcal{N}$ is presented in Figures 2 and 3, along with the outcomes from the ML and L-moment approaches.

In our simulation study, we assess the performance of $\mathcal{N}$ by analyzing its behavior on test sets over a parameter grid of size 20×20 for (σ, ξ) with $\mu = 0$, conducting 100 replications for each configuration. We evaluate the performance of the estimates across different GEV samples of sizes 72, 173, and 1000. In Figures 2 and 3, we compare the logarithm of the root mean squared error (RMSE) of the estimates obtained from the ML method, our neural estimator, and the L-moment method for the parameters σ and ξ. This evaluation is conducted through a surface plot of the logarithm RMSE across the true σ and ξ parameters for varying sample sizes, ranging from small to large. While RMSE serves as a valuable metric for evaluating the accuracy of our neural estimator, it is important to acknowledge its limitations, particularly when dealing with small sample sizes within the ML approach, where it may not necessarily imply accuracy. Although our preference is to use RMSE (MSE), alternative metrics for performance evaluation are also available.

We observe from Figures 2 and 3 that as the sample size increases from sample to large, ranging from 72 to 1000, there is a consistent reduction in RMSE for all approaches. Notably, the neural estimates perform as well as or better than the ML estimates for both σ and ξ, for the small sample sizes 72. As the sample size increases, the MLEs become more stable. It is important to highlight that the RMSE for ξ remains smaller for the neural estimates compared to the MLEs, even as the sample size increases to 173 and 1000. However, in the case of the scale parameter σ, the MLEs demonstrate smaller RMSE values than the neural estimates for larger sample sizes of 173 and 1000. Additionally, it's worth noting that the estimates obtained from the L-moment approach appear less favorable in comparison and have larger RMSE for the estimates compared to our neural estimator $\mathcal{N}$ or the ML approach. We also calculate the GML estimates, and they exhibit similar behavior to the L-moment approach.

Overall, this study provides evidence that the neural estimator can serve as a viable alternative to the traditional approaches for modeling extreme value, particularly the ML or LM approach.

3.2 | Bootstrap

To account for uncertainty in parameter estimates obtained from the neural estimator $\mathcal{N}$, we employ a parametric bootstrap approach (Efron & Tibshirani, 1994). We generate $B = 1000$ bootstrap samples from the original data, estimate the GEV parameters using $\mathcal{N}$ for each sample, and compute simple 95% confidence intervals (CIs) for the true parameters by employing the 2.5th and 97.5th percentiles of the bootstrap estimates. The bootstrap method incurs no additional computational costs after training $\mathcal{N}$, enabling the production of 1000 bootstrap replications and the derivation of results from the NN within seconds.

To evaluate the performance of the bootstrap-based CIs, we compare them to the likelihood-based CIs computed using the standard errors of the MLEs from the Hessian matrix of the ML approach. We restrict the range of ξ to $-0.4 < \xi \leq 0.8$ for stable results, as $\xi > 0.8$ corresponds to highly heavy-tailed GEV distributions, and the results might not be stable for both ML and neural estimator. The `ismev` package, as described in Section 3.1, is used for the computation of CIs for the ML approach. We compute the CIs over a test set with 25 parameter configurations of $\mu \in (1, 50)$, $\sigma \in (0.1, 40)$, and $\xi \in (-0.4, 0.8)$, for a fixed-size GEV-samples (=416), each configuration being replicated 100 times.

Figure 4 presents the ratio of the bootstrap-NN CI widths to the ML-based CI widths across the true parameter values. The boxplot summarizes the spread of this ratio, and our results indicate that the bootstrap-NN CIs are slightly wider than those of the ML. Additionally, in Figure 5, we present the bootstrap coverage probability (BCP) of the neural estimates for ξ, 100-year, and 1000-year return levels. Examining the bootstrap coverage, we find it to be relatively accurate for ξ and the return levels, suggesting reasonable confidence intervals and coverage. However, comparative wider CI widths are observed in the σ estimates obtained from the $\mathcal{N}$, which may be attributed to the reparameterization in our estimation process. Moreover, the CI width obtained from the bootstrap method may exceed that obtained from the ML method because the bootstrap lacks any strong assumption on the distributional constraint.

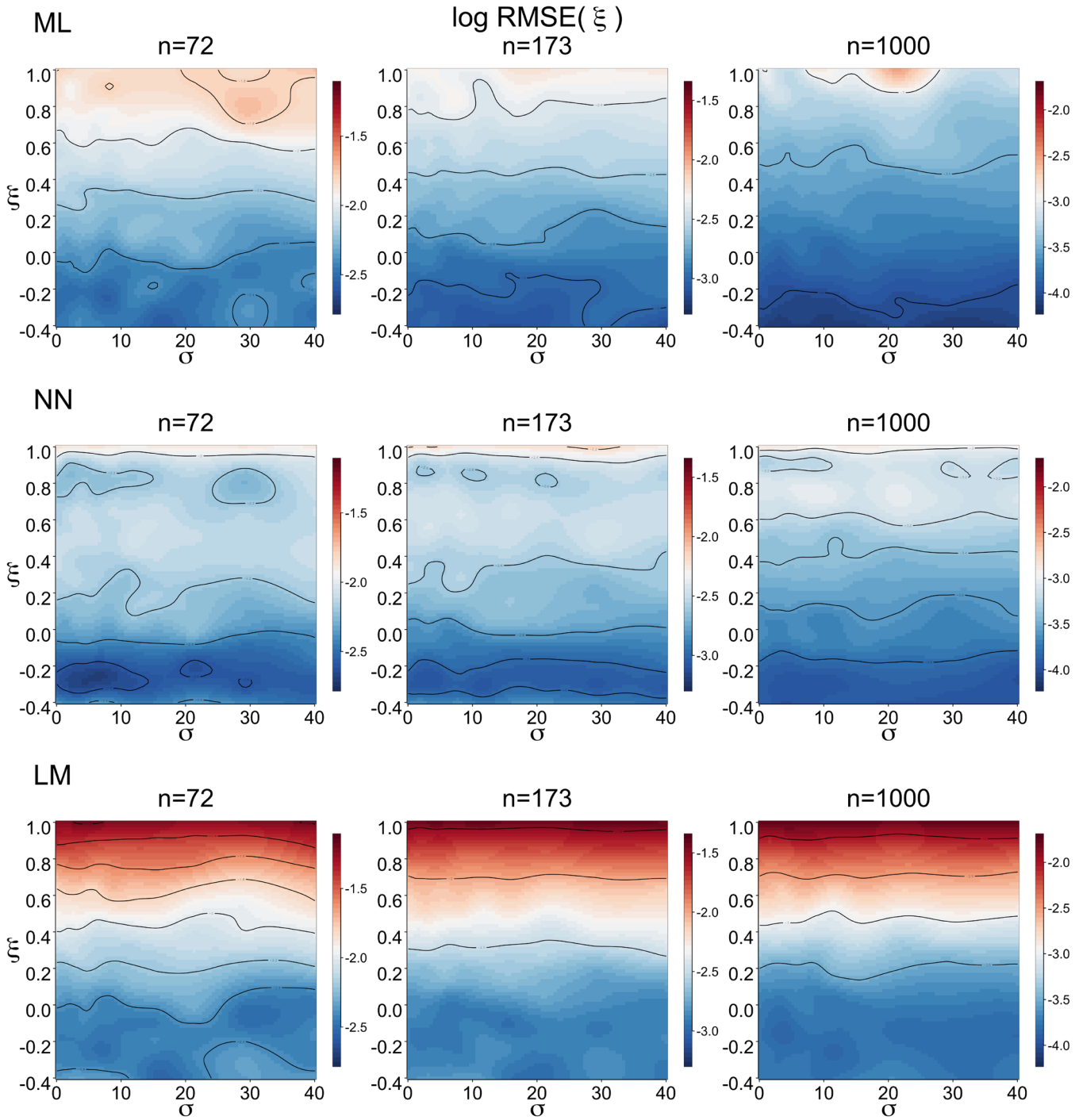

FIGURE 2 This figure illustrates the behavior of the RMSE(ξ) as the sample size varies among 72, 173, 1000 based on our simulation study for the ML, the NN (neural estimator), and the LM approach. The image plot showcases the logarithmic RMSE values concerning the true (σ, ξ), with σ on the x-axis and ξ on the y-axis, displaying the outcomes of each approach in separate rows.

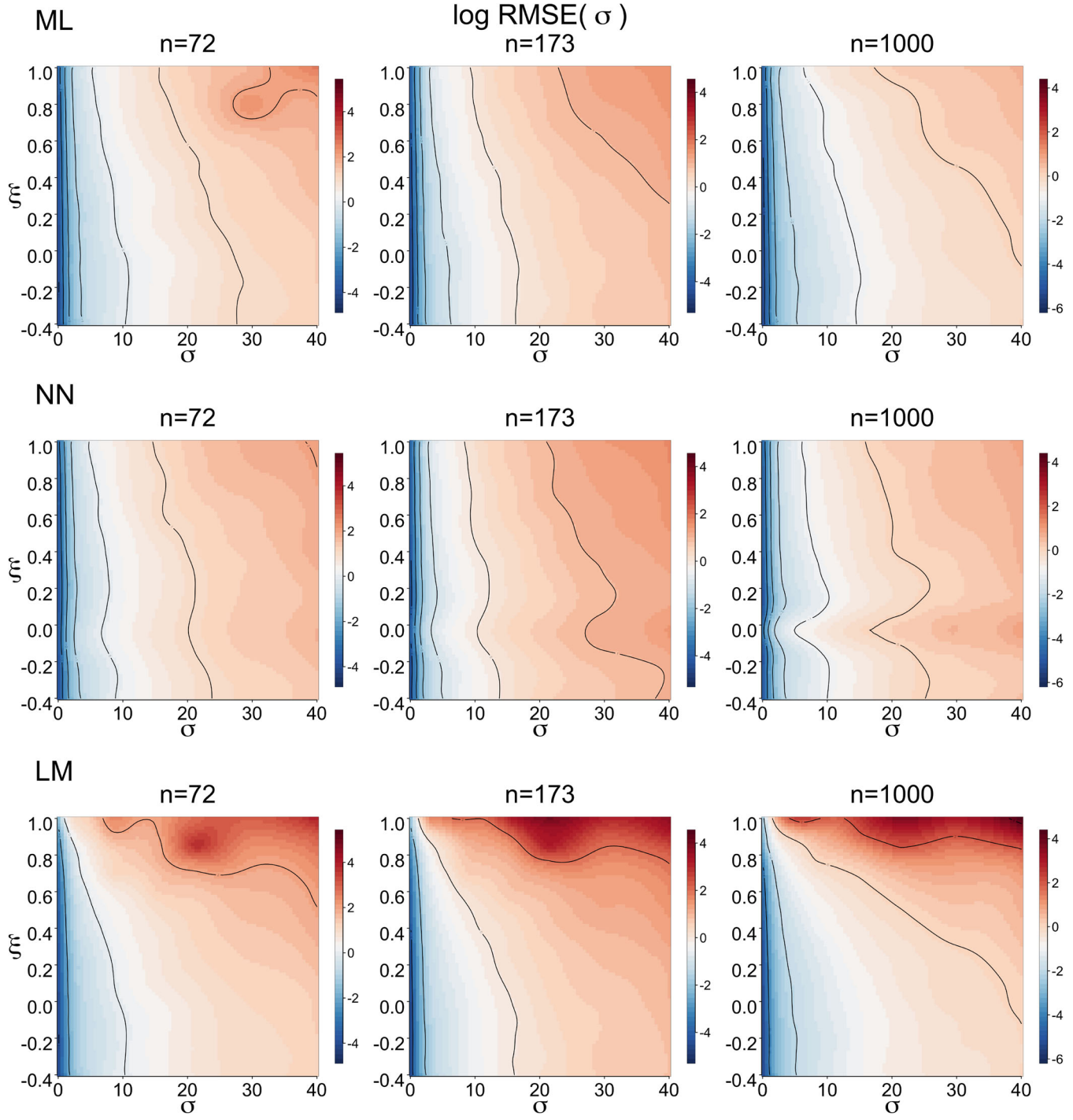

FIGURE 3 This figure illustrates the behavior of the RMSE(σ) as the sample size varies among 72, 173, 1000 based on our simulation study for the ML, the NN (neural estimator), and the LM approach. The image plot showcases the logarithmic RMSE values concerning the true (σ, ξ), with σ on the x-axis and ξ on the y-axis, displaying the outcomes of each approach in separate rows.

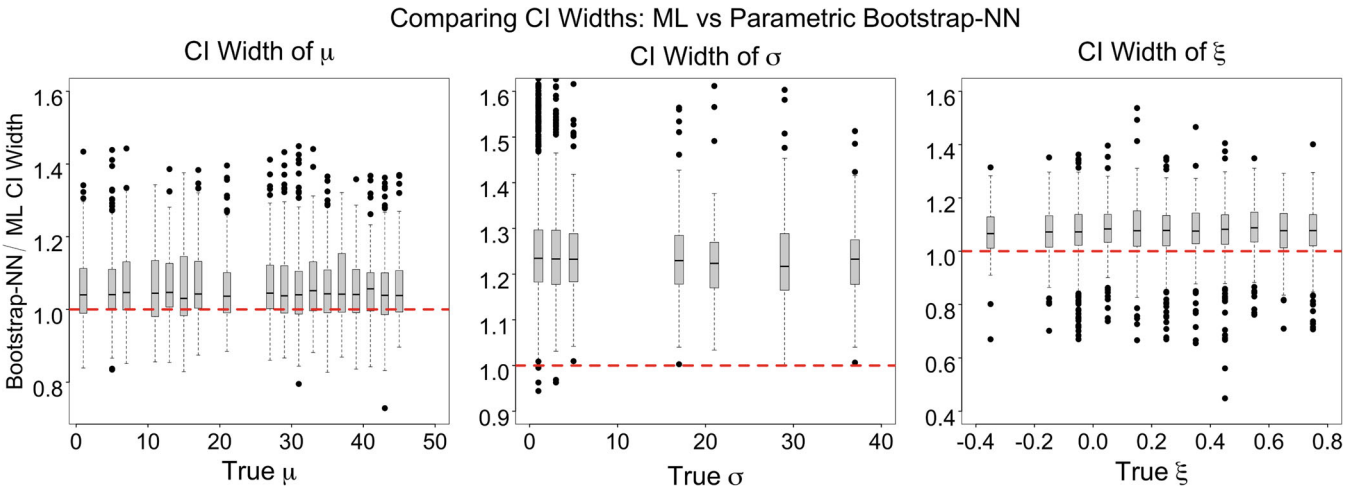

FIGURE 4 The figure compares the CI widths obtained from the Bootstrap-NN and ML approaches for GEV parameters across the true parameter values.

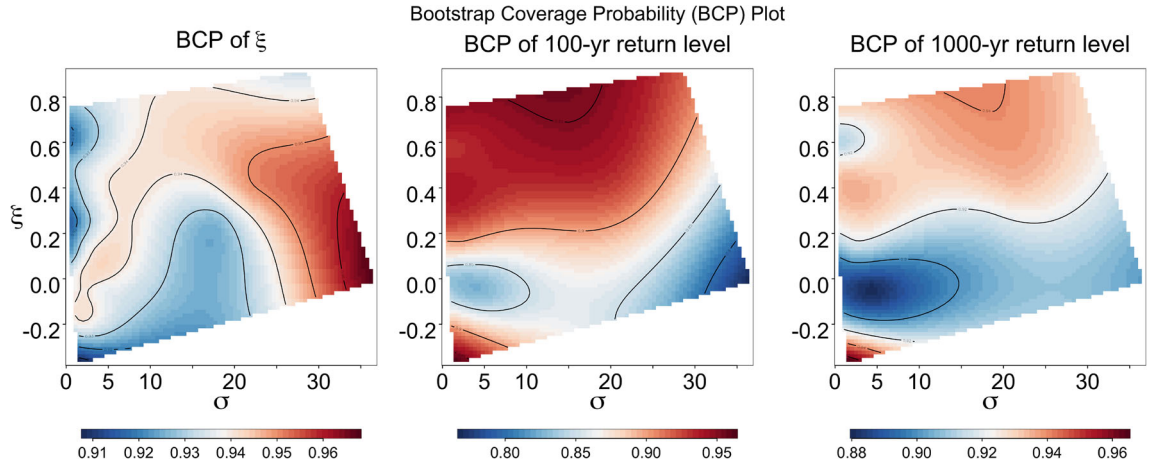

FIGURE 5 Surface plot depicting the bootstrap coverage probability for ξ, the 100-year return level, and the 1000-year return level, plotted against the true σ and ξ parameters.

3.3 | Timing comparison

The NN model has been implemented on the cloud-based Python platform, Google Colab, using a computing environment with 2 virtual CPUs, 32 GB of RAM, and either a P100 GPU with 16 GB of memory or a T4 GPU with 16 GB of GPU memory and expandable system RAM of up to 25 GB. For comparison, MLEs/L-moment estimates have been calculated using the R package `ismev`/`extRemes` on a laptop with a 2.3 GHz Dual Core Intel i5 processor and 8 GB of RAM. The training of the neural model $\mathcal{N}$ takes about 200 s to complete one epoch, and training for 150 epochs with early stopping at the 38th epoch takes a total of 2.11 h. Once $\mathcal{N}$ is trained, it can be used to fit the GEV distribution by simply evaluating the sample to find the parameter estimates. Evaluation times for all approaches on a test set of 10,000 parameter configurations across three different sample sizes (72, 173, and 1000) are reported in Table 3. Based on this, we anticipate a significant speed increase of over 170 times or more in certain cases when scaling up to the target model output for the ML approach.

4 | CASE STUDY

Another approach to validate and assess the timing of our neural estimator $\mathcal{N}$ is to replicate the results from a comprehensive climate model analysis. We examine temperature extremes in the millennial runs of the CCSM3, a globally recognized

TABLE 3 Comparing the evaluation time across different sample sizes.

Estimation approach	72	173	1000
ML	2.92 min	5.81 min	11.66 min
L-moments	22.32 s	27.64 s	1.79 min
Neural estimator	2 s	3 s	4 s

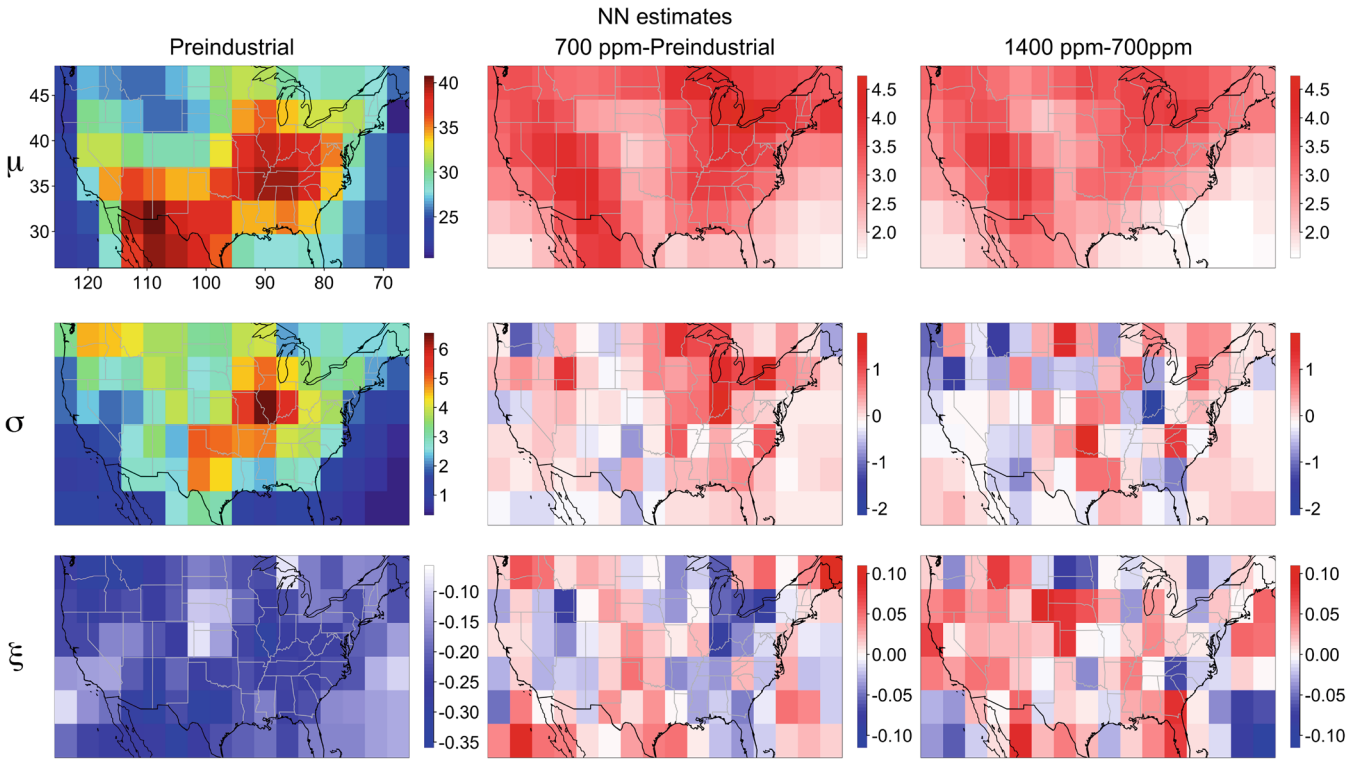

FIGURE 6 NN estimates of CCSM3 GEV parameter for the pre-industrial period and possible changes for future cases. On the left: are estimates for the pre-industrial, center: the expected change in parameter estimates moving from 289 ppm CO_2 to 700 ppm CO_2 concentration, and right: is the change in estimates for 700 ppm CO_2 to 1400 ppm.

climate model, under varying atmospheric CO_2 concentrations (Collins et al., 2006). The CCSM3 model encompasses complete representations of the atmosphere, land, sea ice, and ocean components, operating on grids with T31 resolution for the atmosphere and land, and approximately 1° resolution for the ocean and sea ice (Huang et al., 2016; Yeager et al., 2006). Our study focuses on a 1000-year control run across 133 spatial locations in North America, considering three CO_2 concentration scenarios: pre-industrial (289 ppm), future scenarios with 700 ppm (resulting in a 3.4°C increase in global mean temperature), and 1400 ppm (resulting in a 6.1°C increase in global mean temperature) (Huang et al., 2016). Key external forcings, including solar forcing and aerosol concentrations, are maintained at pre-industrial levels. Each scenario in the climate model undergoes an extensive warm-up period to adjust to forcing changes, enabling the time series of block extremes to become stationary. We calculate the annual maxima of the time series for each scenario to fit the GEV distribution and estimate the changes in maximum temperature on a millennial scale for different CO_2 concentrations. The final 1000-year annual maxima are presumed to achieve stationarity and remain free from climate drift.

We apply the GEV distribution to model the 1000-year annual maximum temperatures across the spatial domain, fitting a specific distribution to each site within the domain for its 1000-year annual maxima values. To estimate the GEV parameters for each grid box, we utilize our trained neural estimator $\mathcal{N}$ as previously described in the sections above. Figure 6 illustrates the neural estimates of CCSM3 GEV parameters for the pre-industrial period and potential changes in future scenarios. Our findings suggest consistency between the GEV distribution for the pre-industrial period and the findings of Huang et al. (2016) concerning μ and ξ, using the ML approach. Negative shape parameters are often observed

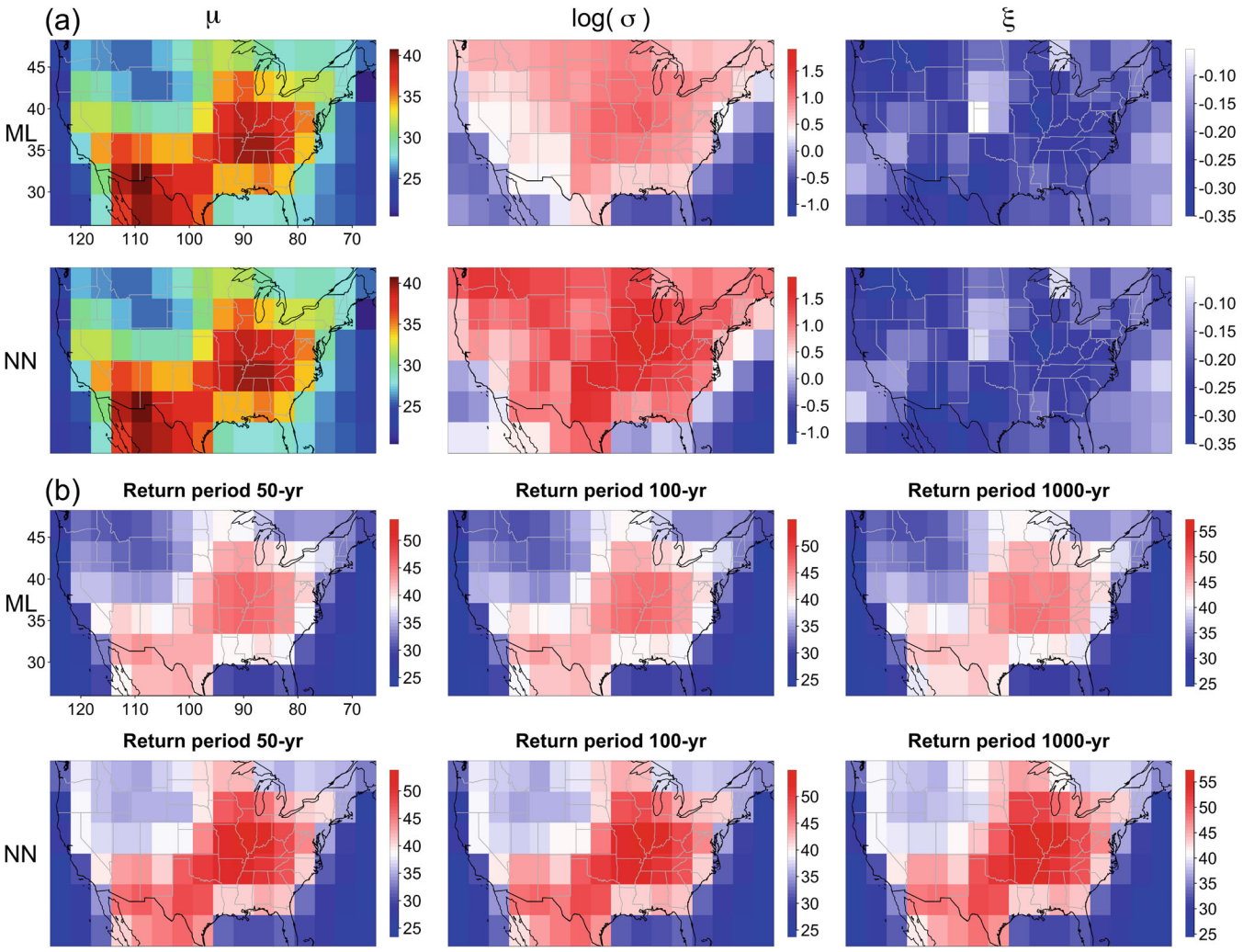

FIGURE 7 (a) Show the CCSM3 GEV parameter estimates from the ML and the neural estimator $\mathcal{N}$ across the 133 spatial locations for pre-industrial concentration scenarios. (b) Comparison plots of return levels for 50-, 100-, and 1000-year periods between the ML approach and $\mathcal{N}$, focusing on pre-industrial concentration scenario.

in extreme temperature modeling, and our model's output validates this trend. We notice an increase in the location parameter μ from the pre-industrial era to both future time series, indicating a mean shift in the GEV distribution, as highlighted in Huang et al. (2016). Additionally, we notice similar changes in σ and ξ from the pre-industrial to 700 ppm scenarios and the transition from 700 to 1400 ppm, which appear as random noise. This may not significantly impact the distribution's scale; however, notable differences in the tail parameter at these levels could be of interest to further explore in the future.

We further provide a focused comparison of the performance of our neural estimator with the likelihood approach in estimating the GEV parameters for the pre-industrial period. It is worth noting that the computational time required to fit the GEV to 133 spatial locations for 1000-year annual maxima using $\mathcal{N}$ is similar to that of the ML estimates, approximately 8 s, although we observe a significant speed-up for larger datasets. Additionally, we present the return levels of the pre-industrial period for both the neural estimator $\mathcal{N}$ and the ML approach for 50-, 100-, and 1000-year periods in Figure 7 for further comparison. Comparable plots for future scenarios (700 ppm and the transition from 700 to 1400 ppm) are accessible in the Appendix, represented in Figures A1 and A2, respectively. Lastly, following the same setup as in Section 3.2, we can calculate a bootstrap-based confidence interval across the spatial domain with 1000 bootstrap replicates in approximately 3 min.

One of the essential applications of building a GEV fitting model for climate models like these is the assessment of the impact of climate change scenarios on extreme temperature events. By employing the GEV distribution and NN

modeling as outlined, we can analyze how changing atmospheric conditions, such as increased CO_2 concentrations, affect the characteristics of extreme temperature events. This analysis provides critical insights into the potential shifts in temperature extremes, aiding in climate adaptation and mitigation strategies. Additionally, the comparison with the traditional ML method allows us to assess the performance and reliability of our neural estimator in capturing these changes accurately. The determination of return levels for different time scales (e.g., 50-, 100-, and 1000-year periods) offers essential information for understanding the changing risk landscape associated with extreme temperatures. Furthermore, the efficient computation of bootstrap-based confidence intervals enhances the robustness of our findings, facilitating more comprehensive climate impact assessments with a reduced computational burden.

5 | CONCLUSION

This study highlights advances in the use of deep learning algorithms for likelihood-free inference. The results indicate that a well-trained NN can estimate the parameters of complex heavy-tailed distributions, such as the GEV, with accuracy comparable to traditional ML approaches. While there may be more variability in estimating the scale parameter using our approach compared to traditional MLE, we believe the estimation of the shape parameter is accurate. Additionally, our findings demonstrate a significant increase in computational speed, with a 170-fold improvement in model evaluation times compared to traditional ML approaches with large datasets. The use of NNs gives us complete control over the testing and training samples, allowing us to operate across a wide range of parameters. This allows us to customize the NN to meet the specific requirements of our problem and assess its reliability.

However, using the NN approach comes with a few limitations. The selection of appropriate hyperparameters is a critical step, that impacts model performance and accuracy. This process can be time-consuming and challenging due to the vast and complex search space of hyperparameters. Optimizing the weights across each layer in the NN can result in a large number of parameters, making the model difficult to manage. Additionally, selecting the appropriate range for the design of the training set is crucial. Careful consideration is needed for choosing informative statistics used as inputs to the network to ensure they provide sufficient information for accurate estimation. It is worth noting that parameter outputs from the NN are often unconstrained, which can be problematic for parameters that should always be within a specific range, such as the scale parameter, which is supposed to be positive at all times. However, by adding constraints to the loss function during training or using reparameterization, the network can avoid problems like these.

Looking ahead, our study opens up several exciting research avenues. Future research opportunities lie in exploring the integration of time-dependent structures into GEV parameter estimation, potentially leading to enhanced accuracy and robustness in extreme value predictions. Additionally, our approach can be extended to develop regression models for the GEV distribution, building upon the pioneering work of Coles (2001). The investigation of its applicability in the context of regression models for the generalized Pareto distribution (GPD), following the foundational work of Davison and Smith (1990) and recent developments by de Carvalho et al. (2021), presents exciting prospects.

Furthermore, as return levels play a significant role in assessing the risk associated with extreme events and predicting future possibilities, the concept of building a neural estimator could be used solely to estimate the return levels for different return periods.

The adaptability of our approach to tasks beyond parameter estimation, such as threshold selection of GPD, holds promise, given its inherent flexibility and computational efficiency. These potential research directions underscore the versatility and significance of our proposed approach within the broader spectrum of extreme value modeling and inference.

Moreover, extending the usage of this approach to spatial modeling of extremes can yield valuable insights into the distribution and behavior of extreme events across diverse geographical locations. Additionally, its application in other statistical modeling approaches related to heavy-tailed distributions can prove beneficial, further enhancing its relevance and impact.

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation Award 2210840 CAS-Climate/Collaborative Research: Prediction and Uncertainty Quantification of Non-Gaussian Spatial Processes with Applications to Large-scale Flooding in Urban Areas. We extend our sincere gratitude to Whitney Huang for generously sharing the 1000 years of output from three multimillennial runs of the CCSM3 model for our case study. We are also grateful to the reviewers and

the associate editor for their thoughtful comments and constructive criticism, which led to significant improvements in the manuscript.

FUNDING INFORMATION

SR, SB, and DWN's work has been partially supported by the National Science Foundation, CMMI-2210840, and SL's work has been partially supported by the National Science Foundation, CMMI-2210811.

DATA AVAILABILITY STATEMENT

All code responsible for generating the results in this article can be accessed through the GitHub repository, available at https://github.com/Sweta-AMS/GEV_NN. The repository contains the necessary code for constructing and training the neural model, evaluating the test set across various sample sizes, and presenting the bootstrapping results. Additionally, we provide the saved trained neural model for evaluation purposes. The repository also features the script file for the Case Study for the Pre-industrial scenario of the CCSM3 data, for demonstration purposes.

ORCID

Soutir Bandyopadhyay https://orcid.org/0000-0003-2213-3333

REFERENCES

Bücher, A., & Segers, J. (2017). On the maximum likelihood estimator for the generalized extreme-value distribution. *Extremes*, *20*, 839–872.

Coles, S. (2001). *An introduction to statistical modeling of extreme values*. Springer.

Coles, S. G., & Dixon, M. J. (1999). Likelihood-based inference for extreme value models. *Extremes*, *2*, 5–23.

Collins, W. D., Bitz, C. M., Blackmon, M. L., Bonan, G. B., Bretherton, C. S., Carton, J. A., Chang, P., Doney, S. C., Hack, J. J., Henderson, T. B., Kiehl, J. T., Large, W. G., McKenna, D. S., Santer, B. D., & Smith, R. D. (2006). The community climate system model version 3 (CCSM3). *Journal of Climate*, *19*(11), 2122–2143.

Cooley, D., Nychka, D., & Naveau, P. (2007). Bayesian spatial modeling of extreme precipitation return levels. *Journal of the American Statistical Association*, *102*(479), 824–840.

Creel, M. (2017). Neural nets for indirect inference. *Econometrics and Statistics*, *2*, 36–49.

Creel, M., & Kristensen, D. (2013). *Indirect likelihood inference (revised)* (UFAE and IAE Working papers 931.13). Unitat de Fonaments de l'Anàlisi Econòmica (UAB) and Institut d'Anàlisi Econòmica (CSIC).

Cremanns, K., & Roos, D. (2017). Deep Gaussian covariance network. arXiv preprint arXiv:1710.06202.

Davison, A. C., Padoan, S. A., & Ribatet, M. (2012). Statistical modeling of spatial extremes. *Statistical Science*, *27*, 161–186.

Davison, A. C., & Smith, R. L. (1990). Models for exceedances over high thresholds. *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, *52*(3), 393–425.

de Carvalho, M., Reis, G., & Kumukova, A. (2021). Regression-type analysis for block maxima on block maxima. arXiv preprint arXiv:2102.09497.

Efron, B., & Tibshirani, R. J. (1994). *An introduction to the bootstrap*. CRC Press.

Engeland, K., Hisdal, H., & Frigessi, A. (2004). Practical extreme value modelling of hydrological floods and droughts: A case study. *Extremes*, *7*, 5–30.

Gamet, P., & Jalbert, J. (2022). A flexible extended generalized pareto distribution for tail estimation. *Environmetrics*, *33*(6), e2744.

Gerber, F., & Nychka, D. (2021). Fast covariance parameter estimation of spatial Gaussian process models using neural networks. *Stat*, *10*(1), e382.

Gilleland, E., & Katz, R. W. (2016). Extremes 2.0: An extreme value analysis package in r. *Journal of Statistical Software*, *72*, 1–39.

Hinton, G., Srivastava, N., & Swersky, K. (2012). Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, *14*(8), 2.

Hosking, J. R. (1985). Algorithm as 215: Maximum-likelihood estimation of the parameters of the generalized extreme-value distribution. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, *34*(3), 301–310.

Hosking, J. R. M., Wallis, J. R., & Wood, E. F. (1985). Estimation of the generalized extreme-value distribution by the method of probability-weighted moments. *Technometrics*, *27*(3), 251–261.

Huang, W. K., Stein, M. L., McInerney, D. J., Sun, S., & Moyer, E. J. (2016). Estimating changes in temperature extremes from millennial-scale climate simulations using generalized extreme value (GEV) distributions. *Advances in Statistical Climatology, Meteorology and Oceanography*, *2*(1), 79–103.

Huser, R., & Wadsworth, J. L. (2022). Advances in statistical modeling of spatial extremes. *Wiley Interdisciplinary Reviews: Computational Statistics*, *14*(1), e1537.

Jenkinson, A. F. (1955). The frequency distribution of the annual maximum (or minimum) values of meteorological elements. *Quarterly Journal of the Royal Meteorological Society*, *81*(348), 158–171.

Jiang, B., Wu, T., Zheng, C., & Wong, W. H. (2017). Learning summary statistic for approximate Bayesian computation via deep neural network. *Statistica Sinica*, *27*, 1595–1618.

Lenzi, A., Bessac, J., Rudi, J., & Stein, M. L. (2023). Neural networks for parameter estimation in intractable models. *Computational Statistics & Data Analysis*, *185*, 107762.

Lenzi, A., & Rue, H. (2023). Towards black-box parameter estimation. arXiv preprint arXiv:2303.15041.

Majumder, R., Reich, B. J., & Shaby, B. A. (2022). Modeling extremal streamflow using deep learning approximations and a flexible spatial process. arXiv preprint arXiv:2208.03344.

Martins, E. S., & Stedinger, J. R. (2000). Generalized maximum-likelihood generalized extreme-value quantile estimators for hydrologic data. *Water Resources Research*, *36*(3), 737–744.

Nelder, J. A., & Wedderburn, R. W. (1972). Generalized linear models. *Journal of the Royal Statistical Society, Series A (Statistics in Society)*, *135*(3), 370–384.

Padoan, S. A., Ribatet, M., & Sisson, S. A. (2010). Likelihood-based inference for max-stable processes. *Journal of the American Statistical Association*, *105*(489), 263–277.

Polson, N. G., & Sokolov, V. (2023). Generative AI for Bayesian computation. arXiv preprint arXiv:2305.14972.

Prescott, P., & Walden, A. (1980). Maximum likelihood estimation of the parameters of the generalized extreme-value distribution. *Biometrika*, *67*(3), 723–724.

Sainsbury-Dale, M., Richards, J., Zammit-Mangion, A., & Huser, R. (2023). Neural bayes estimators for irregular spatial data using graph neural networks. arXiv preprint arXiv:2310.02600.

Sainsbury-Dale, M., Zammit-Mangion, A., & Huser, R. (2023). Likelihood-free parameter estimation with neural bayes estimators. *The American Statistician*, *78*, 1–14.

Singer, S., & Nelder, J. (2009). Nelder-mead algorithm. *Scholarpedia*, *4*(7), 2928.

Smith, R. L. (1985). Maximum likelihood estimation in a class of nonregular cases. *Biometrika*, *72*(1), 67–90.

Stephenson, A., & Heffernan, J. (2014). *ismev: An introduction to statistical modeling of extreme values, original S functions written by Janet E.* Heffernan with R port and R documentation provided by Alec G. Stephenson (R package version 1).

Stojkovic, M., & Simonovic, S. P. (2019). Mixed general extreme value distribution for estimation of future precipitation quantiles using a weighted ensemble-case study of the Lim River Basin (Serbia). *Water Resources Management*, *33*, 2885–2906.

Wang, J., Han, Y., Stein, M. L., Kotamarthi, V. R., & Huang, W. K. (2016). Evaluation of dynamically downscaled extreme temperature using a spatially-aggregated generalized extreme value (GEV) model. *Climate Dynamics*, *47*, 2833–2849.

Wikle, C. K., & Zammit-Mangion, A. (2023). Statistical deep learning for spatial and spatiotemporal data. *Annual Review of Statistics and its Application*, *10*, 247–270.

Yadav, R., Huser, R., & Opitz, T. (2022). A flexible Bayesian hierarchical modeling framework for spatially dependent peaks-over-threshold data. *Spatial Statistics*, *51*, 100672.

Yeager, S. G., Shields, C. A., Large, W. G., & Hack, J. J. (2006). The low-resolution CCSM3. *Journal of Climate*, *19*(11), 2545–2566.

Yoon, S., Cho, W., Heo, J.-H., & Kim, C. E. (2010). A full Bayesian approach to generalized maximum likelihood estimation of generalized extreme value distribution. *Stochastic Environmental Research and Risk Assessment*, *24*, 761–770.

How to cite this article: Rai, S., Hoffman, A., Lahiri, S., Nychka, D. W., Sain, S. R., & Bandyopadhyay, S. (2024). Fast parameter estimation of generalized extreme value distribution using neural networks. *Environmetrics*, *35*(3), e2845. https://doi.org/10.1002/env.2845

APPENDIX A. ADDITIONAL FIGURES

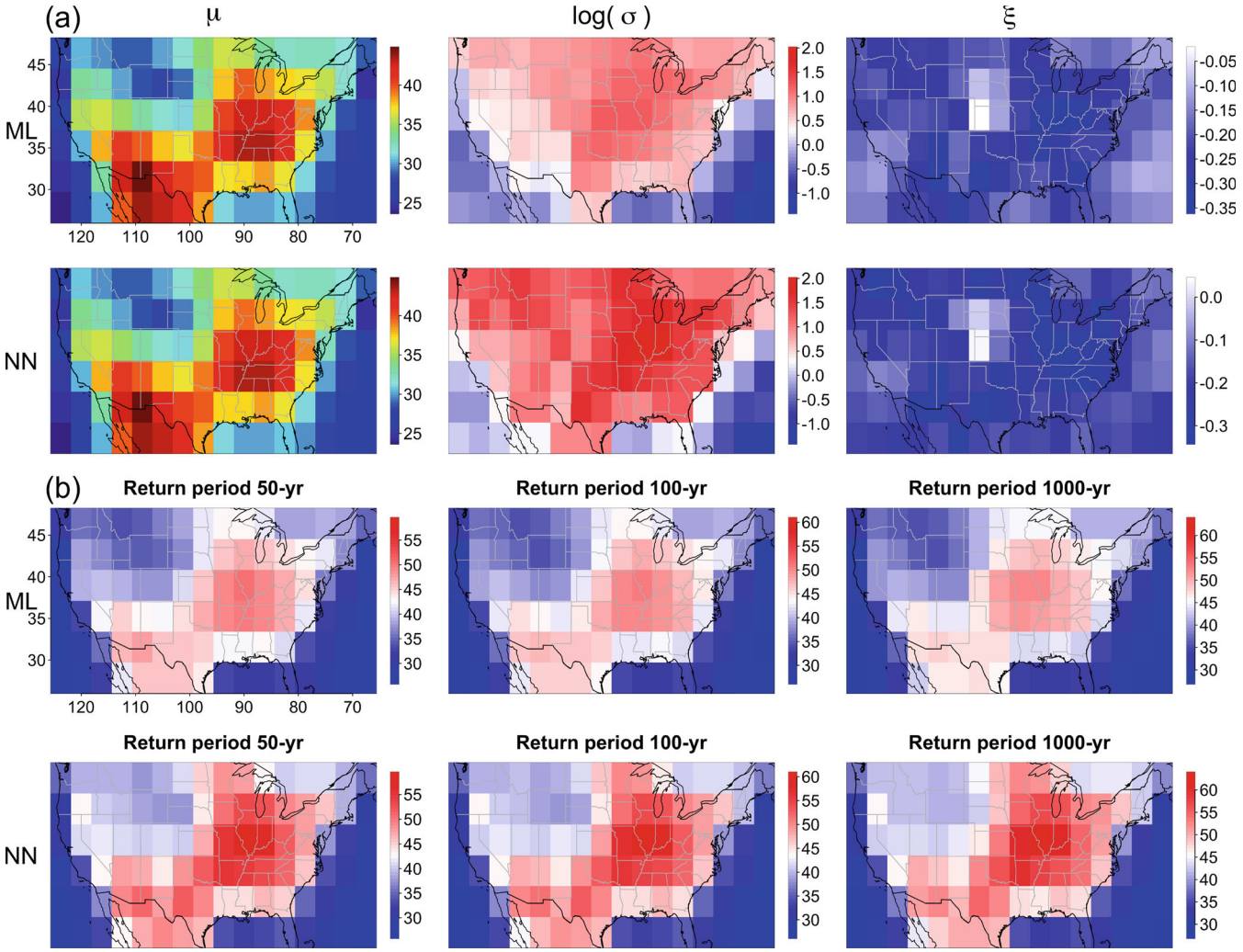

FIGURE A1 (a) Show the CCSM3 GEV parameter estimates from the ML and the neural estimator $\mathcal{N}$ across the 133 spatial locations for the future scenario with 700 ppm (resulting in a 3.4°C increase in global mean temperature). (b) Comparison plots of return levels for 50-, 100-, and 1000-year periods between the ML approach and $\mathcal{N}$, focusing on future scenario with 700 ppm CO_2 concentration.

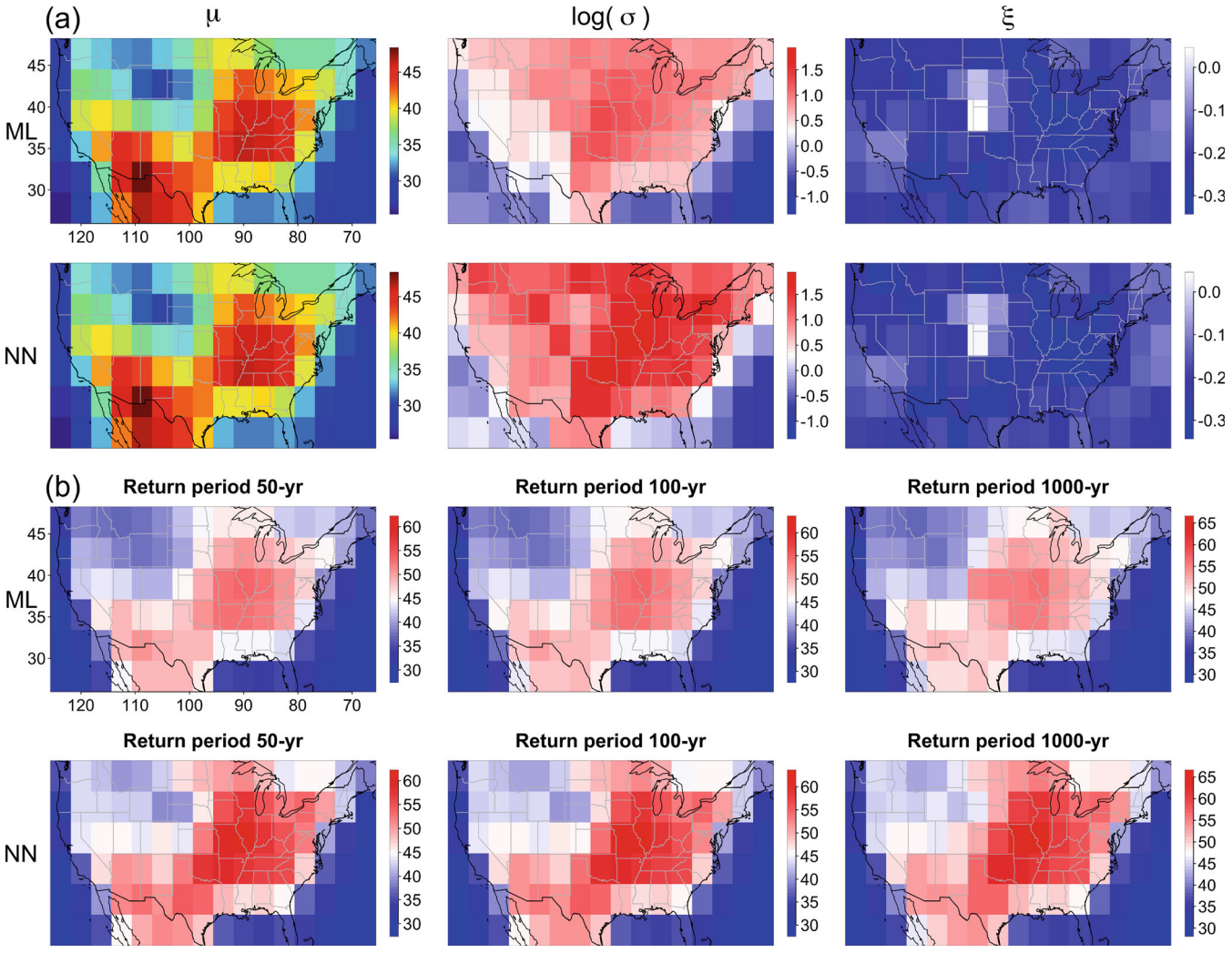

FIGURE A2 (a) Show the CCSM3 GEV parameter estimates from the ML and the neural estimator $\mathcal{N}$ across the 133 spatial locations for the future scenario with 1400 ppm (resulting in a 6.1°C increase in global mean temperature). (b) Comparison plots of return levels for 50-, 100-, and 1000-year periods between the ML approach and $\mathcal{N}$, focusing on future scenario with 1400 ppm CO_2 concentration.