Modeling Extremes with d-max-decreasing Neural Networks

Abstract

We propose a novel neural network architecture that enables non-parametric calibration and generation of multivariate extreme value distributions (MEVs). MEVs arise from Extreme Value Theory (EVT) as the necessary class of models when extrapolating a distributional fit over large spatial and temporal scales based on data observed in intermediate scales. In turn, EVT dictates that d-max-decreasing, a stronger form of convexity, is an essential shape constraint in the characterization of MEVs. As far as we know, our proposed architecture provides the first class of non-parametric estimators for MEVs that preserve these essential shape constraints. We show that our architecture approximates the dependence structure encoded by MEVs at parametric rate. Moreover, we present a new method for sampling high-dimensional MEVs using a generative model. We demonstrate our methodology on a wide range of experimental settings, ranging from environmental sciences to financial mathematics and verify that the structural properties of MEVs are retained compared to existing methods.

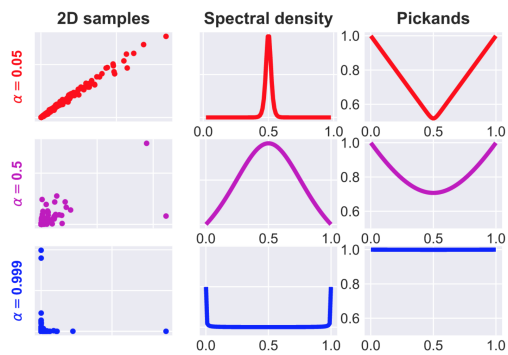


Figure 1: Equivalent representations of MEVs in dimension two, from dependent on the top row to independent at the bottom row. Left column, samples from MEV; Middle column, spectral representation; Right column, Pickands dependence function. We propose methods for estimating the Pickands function (section 2.1), recovering the spectral density (section 2.2) and sampling MEVs (section 4).

1 INTRODUCTION

Modeling the occurrence of extreme events is an important task in many disciplines such as medicine, environmental science, engineering, and finance. For example, understanding the probability of a patient having an adverse reaction to medication or the distribution of economic shocks is critical to mitigating the associated effects of these events [Dey and Yan, 2016]. However, these events are rare in occurrence and therefore are often difficult to characterize with traditional statistical tools. This has been the primary focus of extreme value theory (EVT), which describes how to extrapolate the occurrence of rare events outside the range of available data. In the one-dimensional case, EVT provides remarkably simple models for the asymptotic distribution of the maximum of an infinite number of independent and identically distributed (i.i.d.) random variables, which is due to the celebrated Fisher-Tippet-Gnedenko theorem [Embrechts et al., 1997]. These are known as the generalized extreme value (GEV) distributions [de Haan and Ferreira, 2010].

Perhaps more relevant to practical use-cases is to consider simultaneous extremes in the multi-dimensional scenario. For example, how are extreme weather patterns related in geographical areas or how do extremes of different financial instruments relate? Unlike the one-dimensional case, multivariate extreme value (MEV) distributions generally do not endow simple analytical forms of the underlying density. This leads to difficulties in performing inference tasks using conventional methods. Instead, MEV distributions are characterized by tail dependence functions embedded in extreme value copulas [Pickands, 1981, Gudendorf and Segers, 2010].

Submitted to the 38th Conference on Uncertainty in Artificial Intelligence (UAI 2022). **To be used for reviewing only**.

BACKGROUND: EXTREME VALUE COPULAS

We start with a brief overview of multivariate EVT and provide additional background material in Appendix K. Let Δ_{d-1} denote the unit $d-$dimensional simplex. Let $X_i = (X_1^{(i)}, \ldots, X_d^{(i)}) \in \mathbb{R}^d$ for $i \in \{1, \ldots, n\}$ be a sample of i.i.d. random vectors with common continuous probability distribution F, marginals $F_1, \ldots, F_d$ and copula C_F. Recall that $C_F : [0,1]^d \to [0,1]$ satisfies:

$$C_F(\mathbf{u}) = \mathbb{P}\left[F_1(X_1) \leq u_1, \ldots, F_d(X_d) \leq u_d\right].$$

Let the vector of component-wise maxima be given by: $M^{(n)} = \left(M_1^{(n)}, \ldots, M_d^{(n)}\right)$, where $M_k^{(n)} = \max_{i=1,\ldots,n} X_k^{(i)}$ for $k \in \{1, \ldots, d\}$. Let C_n be the copula of $\bar{M}^{(n)}$ given by: $\bar{M}^{(n)} = \left(\frac{M_1^{(n)} - b_1^{(n)}}{a_1^{(n)}}, \ldots, \frac{M_d^{(n)} - b_d^{(n)}}{a_d^{(n)}}\right)$, where each component-wise maxima $M_k^{(n)}$ is normalized with sequences of real numbers $a_k^{(n)} > 0$ and $b_k^{(n)}$ such that the corresponding limiting marginal is non-degenerate. Then the following property known as max-stability holds:

$$C_n(u_1, \ldots, u_d) = C_F(u_1^{1/n}, \ldots, u_d^{1/n})^n, \ \forall\, \mathbf{u} \in [0,1]^d.$$

We are interested in finding the limiting copula C of C_n as $n \to \infty$. The limiting copula is then called an extreme value copula and we say that C_F is in the *maximum domain of attraction* of C, denoted as $C_F \in \text{MDA}(C)$. The limiting extreme value copula C has the form [Segers, 2012]:

$$C(\mathbf{u}) = \exp\left[\left(\sum_{k=1}^{d} \log u_k\right) A\left(\frac{\log u_1}{\sum_{k=1}^{d} \log u_k}, \ldots, \frac{\log u_d}{\sum_{k=1}^{d} \log u_k}\right)\right], \tag{1}$$

where A is known as a Pickands dependence function that defines the joint dependence of a MEV.

Definition 1 (Pickands dependence function)**.** *A function $A : \Delta_{d-1} \to [1/d, 1]$ is called a Pickands dependence function if it satisfies the following properties:*

1. *A is homogeneous of order 1 and fully d-max-decreasing where d is the dimension;*
2. *A satisfies $\max_{k=1,\ldots,d} w_k \leq A(\mathbf{w}) \leq 1$ for all $\mathbf{w} \in \Delta_{d-1}$.*
3. *$A(\mathbf{e}_k) = 1$ where $\mathbf{e}_k$ is the kth canonical basis vector.*

We give the functional definition of fully d-max-decreasing in Appendix A[1] and instead give the spectral correspondence of A here.

[1] Intuitively, fully d-max-decreasing describes a stronger form of convexity needed to ensure that subsets of margins remain valid MEVs. See Hofmann [2009, Theorem 5.2.2] and Ressel [2013, Theorem 6] for further details.

Definition 2 (Spectral form of Pickands dependence function)**.** *For any Pickands dependence function A, there exists a Borel measure (spectral measure) Λ on Δ_{d-1} satisfying $\int_{\Delta_{d-1}} s_k \, \mathrm{d}\Lambda(\mathbf{s}) = 1$ for $k \in \{1, \ldots, d\}$ such that*

$$A(\mathbf{w}) = \int_{\Delta_{d-1}} \max_{k=1,\ldots,d} w_k s_k \, \mathrm{d}\Lambda(\mathbf{s}), \ \ \mathbf{w} \in \Delta_{d-1}. \tag{2}$$

The equality $\int_{\Delta_{d-1}} s_k \, \mathrm{d}\Lambda(\mathbf{s}) = 1$ is only used as a convention to standardize the margins, and is not essential in maintaining the d-max decreasing property [Fougères et al., 2013]. To provide some intuition on the aims of this paper, Figure 1 illustrates the relationship between different equivalent representations for a canonical parametric MEV – the symmetric logistic distribution with dependence parameter $\alpha = 0.05$ leaning towards complete dependence and $\alpha = 0.999$ leaning towards complete independence. The proposed methods estimates the Pickands function (right most column) and recovers the spectral measure (middle column) which enables sampling MEVs (left most column).

Related Work. A number of techniques have been developed to estimate extreme value copulas from data. The most relevant to the present work is that by Pickands [1981] where a non-parametric estimator of the Pickands function was first proposed. Following works such as Capéraà et al. [1997] and Bücher et al. [2011] describe alternative takes on estimating the dependence function. The above methods, however, do not guarantee that the estimate completely satisfy the conditions of a valid Pickands dependence function. In Marcon et al. [2017], the authors consider a projection of a nonparametric estimator to a convex function represented as a Bernstein polynomial. However, the number of parameters required significantly increases with both the amount of data and the dimensionality, making it difficult for higher dimensional problems or problems with many data points. Finally, a number of Pickands estimators were compared and described in Vettori et al. [2018], and notably none of the estimators reviewed satisfied all requirements of the Pickands function in cases where $d > 2$. For additional details, please refer to the review on extreme value copulas in Gudendorf and Segers [2012]. A theoretical review of d-max-decreasing functions and their applications to copulas is given in Ressel [2019].

Our Contributions.

1. We present fully d-max-decreasing neural networks, an architecture constrained to represent Pickands dependence functions of MEVs.
2. We prove that, in the limit, the proposed architecture can approximate arbitrary Pickands functions.
3. We propose a generative neural network representation of the spectral density of Pickands functions.
4. We propose an extension of the Pickands Estimator to train neural networks.

2 NEURAL REPRESENTATIONS OF EXTREME VALUE DISTRIBUTIONS

Our main results propose two architectures for representing MEVs: a deterministic method for representing the Pickands dependence function, and a stochastic method for representing the spectral measure. While both represent equivalent quantities, each is more suited for a particular task. The deterministic representation is more suitable for estimating exceedance probabilities whereas the spectral representation is more suitable for sample generation.

2.1 FULLY d-MAX-DECREASING NEURAL NETWORKS

We are interested in finding a flexible parameterization of A that enforces all the properties given in Definition 1. The most difficult property to enforce is being fully d-max-decreasing. To that end, we propose a new architecture inspired by Maxout Networks [Goodfellow et al., 2013] and Input Convex Neural Networks (ICNNs) [Amos et al., 2017]. The proposed architecture, dubbed *d-max Neural Networks (dMNNs)*, has additional restrictions to fulfill the conditions of the Pickands dependence function.

Theorem 1 (Fully d-max-decreasing Architecture)**.** *Let $A_{\boldsymbol{\theta}}^{(m)}(\mathbf{w})$ be a function defined as:*

$$A_{\boldsymbol{\theta}}^{(m)}(\mathbf{w}) := \max\left(\max_{k=1,\dots,d} w_k, L^{(m)}(\mathbf{w}) + (1 - L^{(m)}(\mathbf{e})^T\mathbf{w})\right), \tag{3}$$

where

$$L^{(m)}(\mathbf{w}) = \frac{1}{n_m}\sum_{j=1}^{n_m}\left(\ell^{(m)} \circ \ell^{(m-1)} \circ \cdots \circ \ell^{(1)}(\mathbf{w})\right)_j,$$

$$\ell^{(i)}(\mathbf{h}^{(i-1)})_j = \max_{k=1,\dots,n_{i-1}}\left(\Theta^{(i)}_{j,\cdot} \odot h^{(i-1)}\right)_k,$$

$$\mathbf{h}^{(i-1)} = \ell^{(i-1)} \circ \cdots \circ \ell^{(1)}(\mathbf{w}),$$

$$L(\mathbf{e}) = (L(\mathbf{e}_1), \dots, L(\mathbf{e}_d))^T,$$

m is the number of layers, n_i is the width of the ith layer, $\Theta^{(i)} \in \mathbb{R}_+^{n_i \times n_{i-1}}$ are the weights of the ith layer, constrained to be all positive, and $\mathbf{e}_i$ is the ith canonical basis vector. $\odot$ denotes component-wise multiplication.

Then, $A_{\boldsymbol{\theta}}^{(m)}(\mathbf{w})$ is a fully d-max-decreasing function. Moreover, $A_{\boldsymbol{\theta}}^{(m)}(\mathbf{w})$ represents a valid Pickands dependence function.

Intuition of proof. The proof uses the idea that $\mathbb{E}_{\mathbf{s}}[\max_{k=1,\dots,d}(w_k s_k)]$, $\mathbf{s} \in \Delta_{d-1}$ is fully d-max-decreasing and certain compositions of this function retain this property. The full proof is given in Appendix B.1. □

For notational convenience, we drop the (m) unless needed. To get an intuition behind the structure of the architecture, note that in the single layer case in the limit as $n_1 \to \infty$, the weights $\boldsymbol{\theta}$ correspond to samples of the spectral measure in Definition 2 and the expectation is computed empirically. While the proposed architecture is guaranteed to enforce the properties of the Pickands function, and is thus fully d-max-decreasing, we are also interested in seeing how well it can approximate an arbitrary Pickands dependence function. We present results in the following theorem:

Theorem 2 (Uniform Convergence)**.** *Suppose that $\boldsymbol{\theta}$ are samples from the true spectral measure and A is the true Pickands function. The empirical process*

$$\mathbb{G}_n = \sqrt{n}\left(A_{\boldsymbol{\theta}}^{(1)}(\mathbf{w}) - A(\mathbf{w})\right)$$

converges to a zero mean Gaussian process as $n \to \infty$ where $A_{\boldsymbol{\theta}}^{(1)}$ is a single layer dMNN of width n.

Intuition of proof. We first establish pointwise convergence. Then we show A is Lipschitz over a bounded set whose covering number grows in accordance with functions that are $P-$Donsker. The full proof is given in Appendix B.2. □

The result in Theorem 2 has many implications on the properties of the proposed network since it, for example, allows us to quantify the uncertainty associated with our function estimates. Using the proposed architecture, we mitigate issues faced by previous estimators, such as [Bücher et al., 2011, Capéraà et al., 1997, Marcon et al., 2017], in enforcing the d-max-decreasing property, inequalities, and endpoints of the function.

2.2 A GENERATIVE MODEL FOR THE SPECTRAL MEASURE

While the spectral measure can be computed from the weights of the proposed dMNN, we propose an alternative representation of the spectral measure using a generative neural network. We model $\mathbf{y} \sim \Lambda$ in (2) as the output of a generative neural network $G(\,\cdot\,;\boldsymbol{\phi}) \in \mathbb{R}_+^d$ with parameters $\boldsymbol{\phi}$, i.e. $\mathbf{y} = G(\mathbf{z};\boldsymbol{\phi})$ which maps input samples $\mathbf{z} \sim p_z$ to $\mathbf{y}$, where p_z is a distribution that is easy to sample from (such as a multivariate Gaussian distribution). This leads us to a representation of A in terms of the generator:

$$A_G(\mathbf{w}) := \mathbb{E}_{\mathbf{y}\sim G}\left[\max_{k=1,\dots,d} w_k y_k\right], \tag{4}$$

where $\mathbb{E}[y_k] = 1$. The expectation is taken empirically with a large number of samples from G.

Remark 1. *The function given by* (4) *satisfies all the necessary conditions for a valid Pickands function.*

Following Remark 1, we informally note that it follows from the universal approximation theorem of neural networks that if G is sufficiently expressive then (4) can represent an arbitrary Pickands dependence function.

Use Cases of Each Representation. The difference between the representation given by the dMNN (3) and the generative neural network (4) is: in the dMNN case the spectral measure is modeled by a discrete number of elements as dictated by the dMNN architecture, while in the generator case the implicit distribution of the spectral measure is modeled. The dMNN is useful in representing probabilistic quantities since it provides a deterministic representation of the CDF and therefore it does not exhibit the variance of the generative representation. On the other hand, the generative model is capable of simulating many realizations of the MEV, particularly useful for sampling applications.

3 PARAMETER ESTIMATION

Fitting data to high dimensional copulas is often a difficult task since the probability density function (PDF) is not directly modeled. In general, specific parametric families are used to make the process easier, such as in Archimedean copulas. While it is theoretically possible to first obtain the underlying PDF via differentiating the CDF and then fit the dMNN with Maximum Likelihood Estimation (MLE), the procedure is computationally complex, especially in high dimensions. The main drawback of such a method lies in the need to differentiate the $d-$variate CDF, since nested differentiation with existing automatic differentiation methods may result in numerical errors [Margossian, 2019]. Instead, we use specific properties of MEVs to transform the parameter fitting procedure into MLE over univariate random variables. We additionally present the analogs for survival distributions in Appendix C.

3.1 FITTING THE DEPENDENCE FUNCTION

Let F_k denote the univariate marginal CDF (which can be fitted using MLE as in Embrechts et al. [1997] or the L-moments method of Hosking [1990]) of the kth normalized component wise maxima $\bar{M}_k^{(n)} = \frac{M_k^{(n)} - b_k^{(n)}}{a_k^{(n)}}$, $k \in \{1, \dots, d\}$. In addition, let $\mathbf{w} = (w_1, \dots, w_d) \in \Delta_{d-1}$. We introduce the transformation on $\bar{M}_k^{(n)}$:

$$\widetilde{M}_k^{(n)} = -\log(F_k(\bar{M}_k^{(n)})),\ \forall k \in \{1, \dots, d\}, \tag{5}$$

$$Z_w = \min_{k=1,\dots,d} \widetilde{M}_k^{(n)} / w_k. \tag{6}$$

Then, we have: $\mathbb{P}\left[Z_w > z\right] = e^{-zA(\mathbf{w})}$ (for the full derivation, see Section 3 of Gudendorf and Segers [2012]). This transformation casts the original multi-dimensional distribution into the new variables Z_w that are exponentially distributed with rate parameter given by the Pickands dependence function $A(\mathbf{w})$. From this transformation, we can fit the model $A_{\boldsymbol{\theta}}(\mathbf{w})$ to samples Z_w using MLE. This can be done by training the model $A_{\boldsymbol{\theta}}(\mathbf{w})$ with stochastic gradient descent (SGD) to match the data points Z_w as follows:

$$A_{\boldsymbol{\theta}}^{\star}(\mathbf{w}) = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{Z_w} \mathcal{L}(Z_w; \boldsymbol{\theta}), \tag{7}$$

where $\mathcal{L}(Z_w; \boldsymbol{\theta}) = A_{\boldsymbol{\theta}}(\mathbf{w}) Z_w - \log A_{\boldsymbol{\theta}}(\mathbf{w})$. Alternative losses could be considered by reformulating the loss with respect to the estimators defined in Bücher et al. [2011] and Capéraà et al. [1997]. We empirically found that the MLE approach described in (7) provides the best performance, and it follows naturally from the original formulation of Pickands [1981]. The training procedure is summarized in Algorithm 1.

Algorithm 1 Fitting the Pickands-dMNN to Data

Input: $\left\{\left(X_1^{(i)}, \dots, X_d^{(i)}\right)\right\}_{i=1}^{N}$, $N = B \times n$ samples of i.i.d. random vectors where B is the number of blocks of data and n is the size of each block.
Take component-wise maxima over each block: $\left\{\left(M_1^{(n,b)}, \dots, M_d^{(n,b)}\right)\right\}_{b=1}^{B}$ where $M_k^{(n,b)} = \max_{i=(b-1)n+1,\dots,bn} X_k^{(i)}$, $(k, b) \in \{1, \dots, d\} \times \{1, \dots, B\}$.
Fit a GEV to each component-wise maxima $\{M_k^{(n,b)}\}_{b=1}^{B}$, obtain $\{\bar{M}_k^{(n,b)}\}_{b=1}^{B}$, then estimate marginals F_k for each $k \in \{1, \dots, d\}$.
Initialize the parameters $\boldsymbol{\theta} \geq 0$ of the dMNN
Repeat:
Randomly sample a minibatch of training data $\{\bar{M}_k^{(n,b)}\}_{b \in \text{batch}}$ and uniformly sample $\mathbf{w} \in \Delta_{d-1}$.
Transform samples according to Equations (5) and (6) to obtain transformed samples $\{Z_{w,b}\}_{b \in \text{batch}}$.
Compute gradient $\nabla_{\boldsymbol{\theta}} \sum_{b \in \text{batch}} \mathcal{L}\left(Z_{w,b}; \boldsymbol{\theta}\right)$.
Update $\boldsymbol{\theta}$ with Adam [Kingma and Ba, 2014]
Until convergence
Output: $A_{\boldsymbol{\theta}}^{\star}(\mathbf{w})$.

3.2 FITTING THE GENERATOR

Recall that we have an equivalent representation of A given by A_G in (4) where $G(\cdot; \boldsymbol{\phi})$ is a function, with parameters $\boldsymbol{\phi}$, of random variables. We fit the parameters $\boldsymbol{\phi}$ of the generator by solving the following optimization problem:

$$\min_{\boldsymbol{\phi}} \mathbb{E}_{Z_w} \mathcal{L}(Z_w; \boldsymbol{\phi}) + \eta \left\| \mathbb{E}_{\mathbf{y}}[\mathbf{y}] - \mathbf{1}_d \right\|_2^2, \tag{8}$$

with $\mathcal{L}$ now defined using the representation of A_G in (4):

$$\mathcal{L}(Z_w; \boldsymbol{\phi}) = \mathbb{E}_{\mathbf{y}}[\max_{k=1\dots d} y_k w_k] Z_w - \log \mathbb{E}_{\mathbf{y}}[\max_{k=1\dots d} y_k w_k]$$

where $\mathbf{y} = (y_1, \dots, y_d) = G(\mathbf{z}; \boldsymbol{\phi})$, $\mathbf{y} \in \mathbb{R}_+^d$ and $\mathbf{z} \in \mathbb{R}^k \sim p_z$ with $\eta > 0$ as a regularization factor. Note that the second

expectation in (8) is only needed to enforce the margins. It need not be strictly enforced, enforcing approximately only results in minor changes in the tail index. The expectations with respect to $\mathbf{y}$ in (8) are approximated using the sample mean with samples from the generator.

To summarize the parameter estimation section, we bypass the need to differentiate the CDF and use properties of MEVs to estimate the parameters of the distribution from data. Both representations of the Pickands function presented can be used with this technique.

4 SAMPLING

While learning MEV distributions from data is important for computing probabilities, it is also useful to simulate possible scenarios by sampling from an estimated MEV distribution. We introduce a sampling technique using the proposed architectures to efficiently sample from arbitrary MEVs. To the best of our knowledge, there are no general sampling methods for arbitrary extreme value copula that scale to high dimensions. This is because MEV sampling algorithms assume knowledge of the spectral measure, and do not consider sampling when given only the Pickands function. It then becomes necessary to recover the spectral measure from a given Pickands function or from data, which we previously described two methods for doing so. We additionally note that the traditional method of conditional sampling for copulas is ineffective since it requires both computing high order derivatives and using numerical root-finding techniques. We base our sampling procedure on algorithms for the infinite dimensional analogue of MEV distributions known as *max-stable processes* [Dombry et al., 2016]. Max-stable processes have the property that finite dimensional marginals are MEVs and have a spectral representation in terms of the spectral measure Λ for stationary processes. This ultimately allows us to recast MEV sampling in terms of prior work on sampling from max-stable processes, where established methods exist.

4.1 MARGINS OF MAX-STABLE PROCESSES AS MEV DISTRIBUTIONS

A stationary max-stable process has the form:

$$\max_{i\geq 1} \xi_i y_i(x), \;\; x \in \mathbb{X} \subset \mathbb{R}^k \tag{9}$$

where ξ_i is the ith realization of a Poisson point process with intensity $\xi^{-1}\mathrm{d}\xi$. y_i is the ith sample from the spectral measure. Additionally, $\mathbb{E}[y(x)] = 1, x \in \mathbb{X}$ is generally assumed to enforce unit Frechet margins. For a finite number d of $\{x_j\}_{j=1}^d$, this corresponds to a d-dimensional spectral measure with the same properties as in Definition 2. The key idea is to use the representation in (9) to sample from the full MEV distribution with only knowledge of the spectral measure. We use the algorithm mentioned in Hofert et al. [2018, Algorithm 1] for sampling from the full distribution given samples of the spectral measure. We give the details of the algorithm in Appendix I Algorithm 5.

4.2 SAMPLING FROM THE dMNN

Suppose we fit a single layer dMNN using Algorithm 1 with weights given by $\boldsymbol{\theta} \in \mathbb{R}_+^{w\times d}$ where w is the width of the network and d is the data dimension. Consider the transformation $\hat{\theta}_{i,j} = \theta_{i,j}/\sum_{j=1}^d \theta_{i,j}$ where we transform the weights of the network to the unit simplex Δ_{d-1}, and i, j refer to the row and column indices.

We then choose a number N and compute

$$\max_{i=1,\ldots,N} \xi_i \hat{\theta}_{i+j}, \quad j \sim \text{rand}(\{1,\ldots,w-N\})$$

where ξ_i is defined as per (9). While this method is effective in sampling, a possible issue is the finite number of $\hat{\theta}$ dictated by the width w of the network. The generative model on the other hand allows for unlimited generation of samples of the spectral measure.

4.3 SAMPLING FROM THE GENERATIVE MODEL

Suppose we fit a generative model $G(z;\boldsymbol{\phi})$ to data following the optimization procedure in (8). Then sampling proceeds similarly to the case with the dMNN except in this case we do not use the weights of the network explicitly, but sample from the model:

$$\max_{i=1,\ldots,N} \xi_i y_i \;\text{ where }\; y_i = G(z_i;\boldsymbol{\phi}), z_i \sim p(z)$$

where the notation is maintained as above with $p(z)$ defining an easy to sample prior distribution.

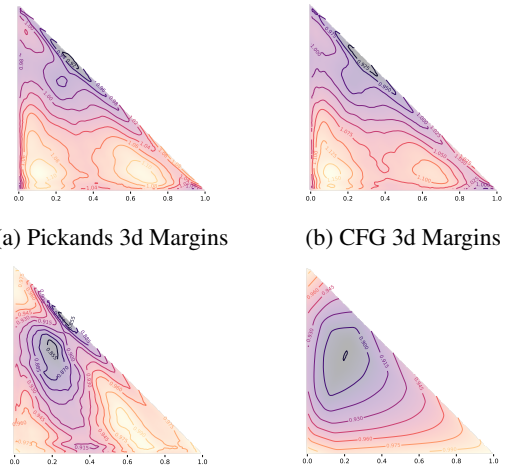
As a final note regarding the sampling methods, one particularly useful way of combining the methods is to first estimate $A_{\boldsymbol{\theta}}$ from data using an estimator such as the dMNN. Then, fit the generator to $A_{\boldsymbol{\theta}}$ by taking the mean squared error (MSE) between the two representations, i.e.

$$\boldsymbol{\phi}^\star = \arg\min_{\boldsymbol{\phi}} \left\| A_{\boldsymbol{\theta}} - \mathbb{E}_{\mathbf{y}\sim G_{\boldsymbol{\phi}}}\left[\max_{k=1,\ldots,d} w_k y_k\right]\right\| + \eta\|\mathbb{E}[\mathbf{y}] - \mathbf{1}_d\|.$$

This provides a simple way to recover the spectral density of any given EVC and thus an effective way to sample from arbitrary MEVs. We detail this algorithm in Appendix I Algorithm 4.

5 RESULTS

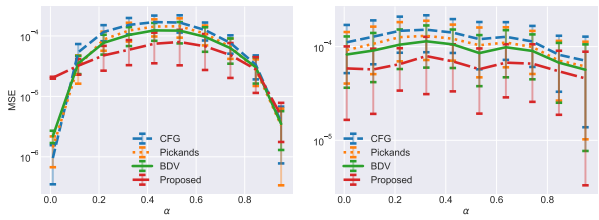
In this section, we provide numerical results that compare the estimation capabilities of the proposed dMNN-



(a) Pickands 3d Margins (b) CFG 3d Margins

(c) BDV 3d Margins (d) dMNN 3d Margins

Figure 2: Qualitative comparison of 3d margins from learned 10d MEV for the commodities dataset. The dMNN retains margins that are valid Pickands dependence function. The other estimators are non-convex and outside the required bounds. Contours plotted with solid lines. See additional figures in Appendix H and D, Figures 20d to 23d.



(a) A_{SL} MSE ($d = 2$) (b) A_{ASL} MSE ($d = 2$)

Figure 3: MSE of survival probabilities for $d = 2$ with 100 samples for A_{SL} (3a) and A_{ASL} (3b). Thresholds are above the 75th percentile.

based model with well-known estimators from the literature: Pickands [Pickands, 1981], CFG [Capéraà et al., 1997], and the estimator described in [Bücher et al., 2011] which we refer to as BDV. These estimators are described in greater detail in Appendix F. We start by evaluating the performance for estimating survival probabilities on known parametric models, followed by real data. We conclude with experiments on sampling from a MEV, where we use the proposed generative model for high dimensional data with different dependence structures. To align with the results in Theorem 2, for the experiments presented in this section, we use a single layer dMNN with a width of 512. Additional experiments with two different architectures are presented in Appendix D.

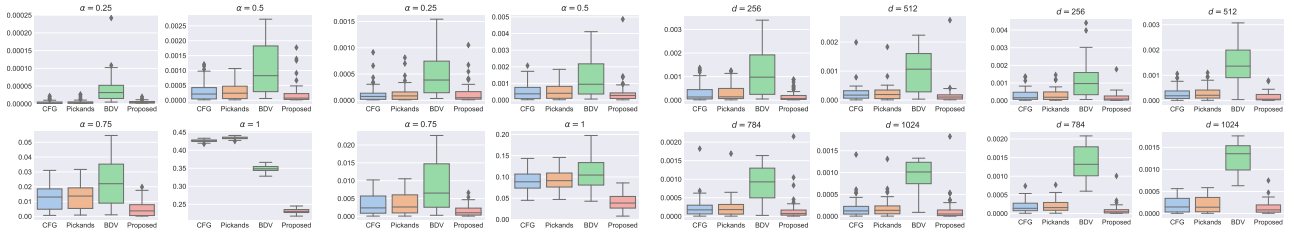
Synthetic data. We consider two canonical families of extreme value distributions known as the symmetric logistic (A_{SL}) and the asymmetric logistic (A_{ASL}) families where

Pickands function	Parameters
$A_{\text{SL}}(\mathbf{w}) = \left(\sum_{k=1}^{d} w_k^{1/\alpha}\right)^{\alpha}$	$\alpha \in (0, 1]$
$A_{\text{ASL}}(\mathbf{w}) = \sum_{b \in \mathcal{P}_d} \left(\sum_{i \in b} (\lambda_{i,b} w_i)^{1/\alpha_b}\right)^{\alpha_b}$	$\alpha_b \in (0, 1]$ $\lambda_{i,b} \in [0, 1]$ $\sum_{i \in b} \lambda_{i,b} = 1$

Table 1: Parametric Pickands functions for the symmetric A_{SL} and asymmetric A_{ASL} logistic copulas and their valid parameter ranges. $\mathcal{P}_d$ refers to the power set of $\{1, \ldots, d\}$. All functions are defined for domain $\mathbf{w} \in \Delta_{d-1}$.

the underlying Pickands function is given by Gudendorf and Segers [2010] listed in Table 1. $\alpha \in (0, 1]$ is the parameter modeling the degree of dependence between variables ranging from complete dependence ($\alpha = 0$) to complete independence ($\alpha = 1$). Exact sampling from distributions of this type are described in Stephenson [2003]. Note that for both the symmetric and asymmetric copulas, the marginals are distributed according to the standard Fréchet distribution. We start by comparing the MSE of survival probabilities for $d = 2$ where the true Pickands dependence function is given by the symmetric or asymmetric model described above for different degrees of dependence α. We compute the exact values of the survival probability and consider survival probabilities associated with margins above the 75th percentile. As shown in Figures 3a and 3b, the proposed Pickands-dMNN estimator achieves the lowest MSE performance for most degrees of dependence α for the symmetric logistic model and all the degrees of the asymmetric logistic model. The proposed method performs worse comparatively in the full dependence case of the symmetric logistic (when all components of the vector are the same) which we suspect is due to difficulties in the optimization procedure of the dMNN. We additionally showcase the ability of the proposed method to model high dimensional extreme value distributions. To do this, we train the Pickands-dMNN with data for $d = 256$ with $\alpha \in \{0.25, 0.50, 0.75, 1.0\}$ and for $d = \{256, 512, 728, 1024\}$ with $\alpha = 0.5$. Then, we compute the MSE between the Pickands-dMNN and the true Pickands function via Monte Carlo with 10,000 uniformly sampled points in Δ_{d-1}. The results are illustrated for varying α in Figures 4a and 4b and for $\alpha = 0.5$ in Figures 4c and 4d. While all hyperparameters were fixed at the beginning and not fine-tuned, we note that performance may improve if additional fine-tuning is performed using a validation set.

Real data. We test the proposed estimator with real data on extreme ozone levels ($d = 4$), wind gusts ($d = 10$), commodity prices ($d = 10$), cryptocurrencies to USD conversion rates ($d = 100$) and S&P 500 components with sufficient history ($d = 418$). We provide details for each dataset in Appendix G. For environmental datasets, we compute the maximum over the different sampling periods, while for the financial data we compute the maximum drawdown. The

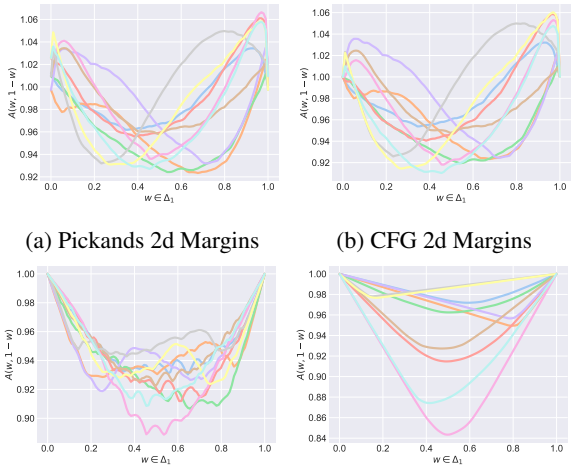


(a) A_{SL} MSE ($d = 256$) (b) A_{ASL} MSE ($d = 256$) (c) A_{SL} MSE ($\alpha = 0.5$) (d) A_{ASL} MSE ($\alpha = 0.5$)

Figure 4: Comparison of $||\hat{A}(\mathbf{w}) - A(\mathbf{w})||_2^2$ for different estimators $\hat{A}$ for different dependence $\alpha = \{0.25, 0.50, 0.75, 1.0\}$ with fixed $d = 256$ (4a, 4b) and for fixed $\alpha = 0.5$ with different $d = \{256, 512, 728, 1024\}$ (4c, 4d). The reference $A(\mathbf{w})$ are A_{SL} (4a, 4c) and A_{ASL} (4b, 4d). Results are over 50 runs with 100 training samples for each run.

	d	Train/Test Length	PICKANDS	CFG	BDV	PROPOSED
Wind	10	day/week	$4.48(18.6)_{\times 10^{-4}}$	*4.15(15.1)*$_{\times 10^{-4}}$	$\mathbf{4.10(16.3)}_{\times 10^{-4}}$	$4.37(17.5)_{\times 10^{-4}}$
Ozone	4	day/week	$3.06(4.66)_{\times 10^{-2}}$	$2.99(4.56)_{\times 10^{-2}}$	*2.86(4.46)*$_{\times 10^{-2}}$	$\mathbf{2.73(4.25)}_{\times 10^{-2}}$
Commodities	10	week/month	$4.34(5.82)_{\times 10^{-3}}$	$4.33(5.71)_{\times 10^{-3}}$	*1.60(1.96)*$_{\times 10^{-3}}$	$\mathbf{1.56(2.21)}_{\times 10^{-3}}$
S&P 500	418	week/month	*3.02(21.2)*$_{\times 10^{-3}}$	*3.02(21.1)*$_{\times 10^{-3}}$	$6.28(35.2)_{\times 10^{-3}}$	$\mathbf{2.41(22.2)}_{\times 10^{-3}}$
Crypto	100	week/month	$1.06(2.85)_{\times 10^{-2}}$	*1.05(4.86)*$_{\times 10^{-2}}$	$1.34(3.44)_{\times 10^{-2}}$	$\mathbf{8.57(26.4)}_{\times 10^{-3}}$

Table 2: MSE of different estimators in estimating maxima over longer time scales. Best and second best performances are marked in **bold** and *italic* respectively.



(a) Pickands 2d Margins (b) CFG 2d Margins

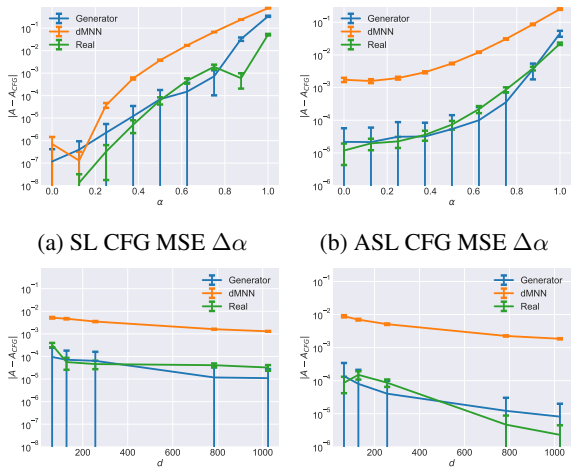
(c) BDV 2d Margins (d) dMNN 2d Margins

Figure 5: Qualitative comparison of 10 out of 45 total 2d margins from learned 10d MEV for the California Winds dataset. The dMNN is the only method that retains margins that are valid Pickands dependence functions. See additional figures in Appendix H and D. Figures 20d- 23d.

maximum drawdown is defined as the difference between the minimum and maximum values over a time period normalized by the maximum value. All margins were fitted with GEVs using the `scipy` implementation.

The main challenge associated with real data is the lack of a ground truth for comparison purposes. It is extremely difficult to accurately compare different estimators on real data because we can never observe the true distribution of extremes. Since the purpose of EVT is to extrapolate to the tails from observations not necessarily in the tails, we consider extreme events on different time scales. If we fit based on extreme observations on shorter time scales and test on extreme observations on longer time scales, we will obtain an estimate of how well the different methods extrapolate to tail probabilities, since longer time scales will have more extreme events.

We compute the accuracy of the different estimators with respect to the empirical estimate on held out data over longer time scales. Specifically, we choose a series of quantiles where we observe data and compute the difference between the estimated survival probabilities and the empirical estimate calculated from observed data. This is quantified as: $\frac{1}{|Q|}\sum_{\gamma \in Q}\left[\frac{1}{B}\sum_{b=1}^{B}\mathbb{1}_{\{M_{n,b} \geq \gamma\}} - P_\theta(M_n \geq \gamma)\right]^2$, where $M_{n,b} = \left(M_{n,b}^{(1)}, \ldots, M_{n,b}^{(d)}\right)$ is the $d-$dimensional vector of point-wise maxima (or point-wise maximum drawdown over a period of interest), P_θ is the estimated survival probability, and Q is a set of thresholds to consider. We choose Q to be all quantiles such that the empirical probability is greater than 0. This measures how well the proposed method can extrapolate to greater extremes over longer time scales. The results are presented in Table 2 and suggest that while most estimators perform similarly, the proposed method most consistently performs the best in terms of the evaluation metric. We would like to emphasize that empirical



(a) SL CFG MSE $\Delta\alpha$ (b) ASL CFG MSE $\Delta\alpha$

(c) SL CFG MSE Δd (d) ASL CFG MSE Δd

Figure 6: MSE of CFG estimate for 1000 samples and 1000 simplex points for $d = 225$ at various $\alpha \in (0, 1)$ (6a, 6b) and for $\alpha = 0.5$ at various $d = \{64, 128, 256, 784, 1024\}$ (6c, 6d). Data sampled from generative model (blue), dMNN (orange), and ground truth (green), where the distributions considered were A_{SL} (6a) and A_{ASL} (6b). Both models were trained with 1000 data points.

evaluation on real data is very challenging, and the high variances prevent us from making meaningful statements on the efficacy of any of the methods. However, from Figures 2d and 5d, we see that our proposed estimator is the only one that satisfies the necessary properties of the Pickands function, which is the main purpose of the proposed method. Additional figures in Appendix H showcase this property on additional datasets and Appendix D Figures 11 to 16 compares these for different architectures. It is critical that these properties are satisfied so that downstream tasks such as conditional probabilities can be computed. From the state of the art estimators, the properties are not satisfied and thus the applicability of the estimators is severely limited.

Sampling from the copula. Finally, to determine the efficacy of sampling from an arbitrary Pickands copula, we consider two synthetic examples using the previously described MEV distributions in Table 1. In this experiment, we train the generator $G(.; \boldsymbol{\phi})$ in (8) based on 1000 samples from the target distribution. We represent the $G(\cdot; \boldsymbol{\phi})$ as a 2 layer 256 width multi-layer perceptron with ReLU activation functions and set $\eta = 1$. Since the Pickands function completely determines the dependency of the random variables, we compare the CFG estimate of the Pickands function from generated samples to the true Pickands function as a measure of sampling quality. We use the CFG estimator due to its ubiquity in the literature and its highly regarded status as a standard estimator for the Pickands dependence function. The results for generating 225 dimensional samples with varying dependence $\alpha \in [0, 1]$ are shown in Figures 6a and 6b. The figures suggest that the generative model performs comparatively well for both distributions considered, with the worst performance occurring in the nearly independent cases ($\alpha = 1$). This is expected, since independence implies a spectral measure with delta functions on the corners of the simplex, which is difficult to learn (see the bottom row of Figure 1 as an example). The figures additionally suggest that sampling using the learned weights of the dMNN has lower variance (since the spectral measure in this is a finite discrete approximation) but does not perform as well in sampling as the generative model. The error of the CFG estimate for the proposed sampling methods (blue and orange) and the exact sampling (green) follow very similar trends in errors, suggesting that both sampling methods are recovering the true spectral measure.

6 CONCLUDING REMARKS

We introduced a new neural network architecture for modeling MEV distributions while enforcing all the properties of the distribution. We additionally show that the architecture can approximate any Pickands function, which allows for precise representations of MEV distributions. Finally, we present a generative model for recovering the spectral representation. Numerical results are provided to empirically demonstrate the effectiveness of the methods in their respective tasks. However, there are some limitations of the proposed methods.

Limitations of Pickands-dMNNs and Generative Model. The main challenge associated with modeling using dMNNs are optimization and architectural choices. Choosing appropriate hyperparameters is a difficult and opaque task that requires additional care. This is a case where non-parametric methods are advantageous, at the cost of being unable to guarantee the necessary properties of the function. Additional progress on understanding the training deep neural networks should improve the representational capabilities of the dMNNs, given its theoretical potential to approximate any Pickands functions to arbitrary precision. Optimization of the generative model suffers from the same issues.

Future Work. The proposed methods have possible applications in a variety of modeling situations. One possibility is to use the dMNN to compute conditional probabilities, which is useful in a variety of classification tasks. Another is in using the spectral measure for finding groups of variables that are extreme simultaneously, such as in [Engelke and Ivanovs, 2021]. Finally, applications of extremes are important in understanding robustness properties of neural networks [Weng et al., 2018], and the proposed work provides foundation for high dimensional extensions.

References

Brandon Amos, Lei Xu, and J Zico Kolter. Input convex neural networks. In *International Conference on Machine Learning*, pages 146–155. PMLR, 2017.

Axel Bücher, Holger Dette, Stanislav Volgushev, et al. New estimators of the pickands dependence function and a test for extreme-value dependence. *The Annals of Statistics*, 39(4):1963–2006, 2011.

Philippe Capéraà, A-L Fougères, and Christian Genest. A nonparametric estimation procedure for bivariate extreme value copulas. *Biometrika*, 84(3):567–577, 1997.

Laurens de Haan and Ana Ferreira. *Extreme Value Theory: An Introduction (Springer Series in Operations Research and Financial Engineering)*. Springer, 2010.

Laurens De Haan et al. A spectral representation for max-stable processes. *The annals of probability*, 12(4):1194–1204, 1984.

Dipak K Dey and Jun Yan. *Extreme value modeling and risk analysis: methods and applications*. CRC Press, 2016.

Clément Dombry, Sebastian Engelke, and Marco Oesting. Exact simulation of max-stable processes. *Biometrika*, 103(2):303–317, 2016.

Paul Embrechts, Thomas Mikosch, and Claudia Klüppelberg. *Modelling Extremal Events: For Insurance and Finance*. Springer-Verlag, Berlin, Heidelberg, 1997. ISBN 3540609318.

Sebastian Engelke and Jevgenijs Ivanovs. Sparse structures for multivariate extremes. *Annual Review of Statistics and Its Application*, 8:241–270, 2021.

Anne-Laure Fougères, Cécile Mercadier, and John P Nolan. Dense classes of multivariate extreme value distributions. *Journal of Multivariate Analysis*, 116:109–129, 2013.

Ian Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. In *International conference on machine learning*, pages 1319–1327. PMLR, 2013.

Gordon Gudendorf and Johan Segers. Extreme-value copulas. In *Copula Theory and Its Applications*. Springer Berlin Heidelberg, 2010.

Gordon Gudendorf and Johan Segers. Nonparametric estimation of an extreme-value copula in arbitrary dimensions. *Journal of multivariate analysis*, 102(1):37–47, 2011.

Gordon Gudendorf and Johan Segers. Nonparametric estimation of multivariate extreme-value copulas. *Journal of Statistical Planning and Inference*, 142(12):3073 – 3085, 2012. ISSN 0378-3758.

Marius Hofert, Raphaël Huser, and Avinash Prasad. Hierarchical archimax copulas. *Journal of Multivariate Analysis*, 167:195–211, 2018.

Daniel Hofmann. *Characterization of the D-norm corresponding to a multivariate extreme value distribution*. PhD thesis, Universität Würzburg, 2009.

J. R. M. Hosking. L-moments: Analysis and estimation of distributions using linear combinations of order statistics. *Journal of the Royal Statistical Society. Series B (Methodological)*, 52(1):105–124, 1990.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *http://arxiv.org/abs/1412.6980*, 2014.

Giulia Marcon, SA Padoan, Philippe Naveau, Pietro Muliere, and Johan Segers. Multivariate nonparametric estimation of the pickands dependence function using bernstein polynomials. *Journal of Statistical Planning and Inference*, 183:1–17, 2017.

Charles C Margossian. A review of automatic differentiation and its efficient implementation. *Wiley interdisciplinary reviews: data mining and knowledge discovery*, 9 (4):e1305, 2019.

J Pickands. Multivariate extreme value distributions, bull. int. statist. 1981.

Paul Ressel. Homogeneous distributions—and a spectral representation of classical mean values and stable tail dependence functions. *Journal of Multivariate Analysis*, 117:246–256, 2013.

Paul Ressel. Copulas, stable tail dependence functions, and multivariate monotonicity. *Dependence modeling*, 7(1): 247–258, 2019.

Johan Segers. Max-stable models for multivariate extremes. *REVSTAT*, 10:61–82, 2012.

Bodhisattva Sen. A gentle introduction to empirical process theory and applications. *Lecture Notes, Columbia University*, 2018.

Alec Stephenson. Simulating multivariate extreme value distributions of logistic type. *Extremes*, 6(1):49–59, 2003.

Sabrina Vettori, Raphaël Huser, and Marc G Genton. A comparison of dependence function estimators in multivariate extremes. *Statistics and Computing*, 28(3):525–538, 2018.

Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. Evaluating the robustness of neural networks: An extreme value theory approach. *arXiv preprint arXiv:1801.10578*, 2018.

Appendix

A FULLY d-MAX-DECREASING FUNCTIONS

We use the definition given in Hofmann [2009]. A function $A(\mathbf{w}) : \mathbb{R}^d \to \mathbb{R}$ is fully d-max-decreasing if and only if for any $\mathbf{x} \leq \mathbf{y} \leq 0$ and any subset $E \subsetneq \{1, \ldots, d\}$:

$$\sum_{\substack{\mathbf{m}=\{0,1\}^d, \\ m_j=1 \text{ if } j\in E}} \left[(-1)^{d+1-\sum_{j\leq d} m_j} \left(\sum_{j\leq d} -y_j^{m_j} x_j^{1-m_j} \right) A\left(\frac{y_1^{m_1} x_1^{1-m_1}}{\sum_{j\leq d} y_j^{m_j} x_j^{1-m_j}}, \cdots, \frac{y_d^{m_d} x_d^{1-m_d}}{\sum_{j\leq d} y_j^{m_j} x_j^{1-m_j}} \right) \right] \geq 0,$$

and $A(\mathbf{e}_i) = 1$ where $\mathbf{e}_i$ is the canonical basis function. Moreover, from Hofmann [2009, Theorem 3.1.1], the following three characterizations are equivalent:

1. The function
$$\exp\left(- \sum_{j=1,\ldots,d} x_j A\left(\frac{x_1}{\sum_{j=1,\ldots,d} x_j}, \cdots, \frac{x_d}{\sum_{j=1,\ldots,d} x_j} \right) \right)$$
defines a multivariate extreme value distribution;
2. There exists a spectral measure Λ such that
$$A(\mathbf{w}) = \int_{\Delta_{d-1}} \max_{k=1,\ldots,d} w_k s_i \, \mathrm{d}\Lambda(\mathbf{s}), \quad \mathbf{w} \in \Delta_{d-1};$$
3. $A(\mathbf{w})$ is fully d-max-decreasing.

Next, we note the nesting property of d-max-decreasing functions, given in Hofert et al. [2018, Section 2.2], that hierarchies of spectral measures define valid EVDs, i.e.

$$A(\mathbf{w}) = \mathbb{E}\left[s_1^{(2,1)} \mathbb{E}\left[\max_{k=1,\ldots,d} s_k^{(1,1)} w_k \right], \ldots, s_d^{(2,1)} \mathbb{E}\left[\max_{k=1,\ldots,d} s_k^{(1,d)} w_k \right] \right].$$

We use part of this property in the next section to define the d-max neural network.

B PROOFS

B.1 d-MAX-DECREASING NEURAL NETWORKS

We partition the proof into the 1-layer case and the n-layer case. We assume that all parameters $\theta \in [0, 1]$ for the purposes of the proof.

Background: $D-$norms. $D-$norms are norms defined as

$$\|\mathbf{x}\|_D := d\,\mathbb{E}_{\boldsymbol{\theta}}\left[\max_{k=1\dots d}(|x_k|\theta_k)\right], \quad \mathbf{x}\in\mathcal{X}\subset\mathbb{R}^d, \tag{10}$$

where $\boldsymbol{\theta}\in[0,1]^d$ and $\mathbb{E}[\theta_k]=1/d$, for $i=1\dots d$. We note that the condition $\mathbb{E}[\theta_k]=1/d$ is not necessary for the $d-\max$ decreasing property, though it leads to unit exponential margins for convenience during inference, see Fougères et al. [2013, Section 2.1] or Hofmann [2009, Definition 5.4.1] for more on this property. The key condition is that the expectation in (10) is taken with respect to the distribution of $\boldsymbol{\theta}$ which has support only on nonnegative real numbers. Taking $\mathcal{X}$ to be the unit simplex, we see that a $D-$norm defines a Pickands dependence function, and by the spectral representation of the Pickands function, all Pickands functions are $D-$norms. The main property we will use throughout the proof is that compositions of $D-$norms are also $D-$norms. This property is well established in, for example, Hofert et al. [2018] and Hofmann [2009]. We finally note that all $D-$norms satisfy the fully $d-$max decreasing property defined as shown in Hofmann [2009, Theorem 3.1.1] .

1-Layer Case.

Proof. Recall that the 1-layer dMNN is given by

$$A_\theta^{(1)}(\mathbf{w}) = \max\left(L^{(1)}(\mathbf{w}) + (1-L^{(1)}(\mathbf{e})^T\mathbf{w}), \max_{k=1\dots d} w_k\right), \quad \mathbf{w}\in\Delta_{d-1} \tag{11}$$

$$L^{(1)}(\mathbf{w}) = \frac{1}{n_1}\sum_{j=1}^{n_1}\left(\max_{k=1\dots d} w_k\theta_{kj}\right)_j \tag{12}$$

The expression (12), corresponding to the first term in (11), is a valid $D-$norm since it is the expectation with respect to a nonnegative spectral measure. The second term $(1-L^{(1)}(\mathbf{e})^T\mathbf{w}) = \sum_{i=1}^d(1-\mathbb{E}[\theta_i])w_i$ is also a $D-$norm since $(1-\mathbb{E}[\theta_i])$ is positive, for $i=1\dots d$, and thus it is also an expectation with respect to a nonnegative spectral measure. In fact, it is equivalent to $\|\text{diag}(1-\mathbb{E}[\boldsymbol{\theta}])\mathbf{w}\|_1$. The combination $L^{(1)}(\mathbf{w}) + (1-L^{(1)}(\mathbf{e})^T\mathbf{w})$ is then the sum of two $D-$norms, equivalent to a composition with the $\|\cdot\|_1$ norm, which is again a $D-$norm. Finally, the outer $\max$ with $\max_{i=1\dots d} w_i$, a $D-$norm corresponding to dependence, is yet another composition of $D-$norms. This results in a function that is fully $d-\max$ decreasing and concludes the proof for the single layer case. □

n-Layer Case.

Proof. We first show the base case (the 2-layer case), then show that the general n-layer case follows. We focus on the composition of intermediate layers, since the technique for proving the output layer is a $D-$norm follows from the 1-layer case. Recall that the 2-layer dMNN is given by

$$A_\theta^{(2)}(\mathbf{w}) = \max\left(L^{(2)}(\mathbf{w}) + (1-L^{(2)}(\mathbf{e})^T\mathbf{w}), \max_{k=1\dots d} w_k\right), \quad \mathbf{w}\in\Delta_{d-1}$$

$$L^{(2)}(\mathbf{w}) = \frac{1}{n_2}\sum_{j=1}^{n_2}\left(\ell^{(2)}\left(\ell^{(1)}(\mathbf{w})\right)\right)_j$$

Let $\ell^{(1)}(\mathbf{w})$ have width n_1. Then the output of $\ell^{(1)}$ is given by the following vector

$$\ell^{(1)}(\mathbf{w}) = \begin{pmatrix} \mathbb{E}_{\boldsymbol{\theta}^{(1,1)}\sim\lambda^{(1,1)}}\left[\max_{k=1\dots d}\theta_k^{(1,1)}w_k\right] \\ \vdots \\ \mathbb{E}_{\boldsymbol{\theta}^{(1,n_1)}\sim\lambda^{(1,n_1)}}\left[\max_{k=1\dots d}\theta_k^{(1,n_1)}w_k\right] \end{pmatrix}, \quad \boldsymbol{\theta}\in[0,1]^d. \tag{13}$$

Each row in (13) is a $D-$norm where the expectation is taken over a delta function centered at $\boldsymbol{\theta}$, i.e. $\lambda^{(1,j)} = \delta(\boldsymbol{\theta}^{(1,j)})$ for $j=1,\dots,n_1$. Therefore, the property of $D-$norms is preserved for each row of (13). By analogy to $\ell^{(1)}$ in (13), the property of $D-$norms is preserved for $\ell^{(2)}$ in (14):

$$\ell^{(2)}(\mathbf{w}) = \begin{pmatrix} \mathbb{E}_{\boldsymbol{\theta}^{(2,1)}\sim\lambda^{(2,1)}}\left[\max_{k=1\dots n_1}\theta_k^{(2,1)}w_k\right] \\ \vdots \\ \mathbb{E}_{\boldsymbol{\theta}^{(2,n_2)}\sim\lambda^{(2,n_2)}}\left[\max_{k=1\dots n_1}\theta_k^{(2,n_2)}w_k\right] \end{pmatrix}, \quad \boldsymbol{\theta}\in[0,1]^d. \tag{14}$$

We then use the nesting property of $D-$norms given in Hofert et al. [2018] such that $\ell^{(2)}(\ell^{(1)}(\boldsymbol{w}))$ is a $D-$norm, and by the same construction, $\ell^{(n)}(\ell^{(n-1)}(\cdots(\ell^{(1)}(\boldsymbol{w}))))$ is a $D-$norm and is thus fully $d-\max$-decreasing. Following the arguments in the 1-layer case for the output layer then completes the proof. □

B.2 UNIVERSAL APPROXIMATION

Our proof that our architecture is an universal approximator of Pickand's copula functions is constructive. Recall that every Pickands function has the form

$$A(\mathbf{w}) = \mathbb{E}_{\mathbf{s}\sim\lambda(\Delta_{d-1})}\left[\max_{k=1\ldots d} s_k w_k\right], \quad \mathbf{w} \in \Delta_{d-1}, \tag{15}$$

where λ is a spectral measure with $\text{supp}(\lambda) = \Delta_{d-1}$. We now construct a single layer dMNN, with width n, by sampling n independent and identically distributed (i.i.d.) samples $\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(n)} \sim \lambda(\Delta_{d-1})$, and setting

$$\tilde{A}_n(\mathbf{w}) = \frac{1}{n}\sum_{j=1}^{n} \max_{k=1\ldots d}(s_k^{(j)} w_k) \tag{16}$$

Before showing that $\tilde{A}_n$ converges uniformly to A, we show that it converges point-wise. Although this intermediary result is not needed to show uniform converge, its proof provides intuition while being less technical.

The copula $\tilde{A}_n$ converges pointwise to A, almost surely.

Proof. Consider the discrete distribution $\mathbb{\Lambda}_n$ given by the n i.i.d. samples $\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(n)} \sim \lambda(\Delta_{d-1})$:

$$\mathbb{\Lambda}_n := \frac{1}{n}\sum_{i=1}^{n} \delta(\mathbf{s}^{(i)}),$$

where $\delta(\mathbf{s})$ represents a Dirac measure at $\mathbf{s}$. By the law of large numbers, for every $\mathbf{w} \in \Delta_{d-1}$,

$$\begin{aligned} \mathbb{E}_{\mathbf{s}\sim\mathbb{\Lambda}_n}\left[\max_{k=1\ldots d} s_k w_k\right] &\xrightarrow{a.s.} \mathbb{E}_{\mathbf{s}\sim\lambda}\left[\max_{k=1\ldots d} s_k w_k\right], \quad n \to \infty \\ \implies A_{\boldsymbol{\theta}}(\mathbf{w}) &\xrightarrow{a.s.} A(\mathbf{w}). \end{aligned}$$

□

We now state and prove the main result regarding uniform convergence.

The empirical process

$$\mathbb{G}_n = \sqrt{n}\left(\tilde{A}_n - A\right)$$

weakly converges to a zero-mean Gaussian process as $n \to \infty$ where $\tilde{A}_n$ is a single layer dMNN with width n.

Proof. Let λ be the law given by the spectral measure $\lambda(\Delta_{d-1})$ and the discrete empirical spectral measure be given by $\mathbb{\Lambda}_n := \frac{1}{n}\sum_{j=1}^{n} \delta(\mathbf{s}^{(j)})$ for n i.i.d. samples $\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(n)}$ from λ. We additionally write $\lambda f := \mathbb{E}_{\mathbf{s}\sim\lambda}[f(\mathbf{s})]$ as the expectation with respect to the measure λ. The empirical process $\mathbb{G}_n$ is defined by

$$\begin{aligned} \mathbb{G}_n &= \sqrt{n}\left(\tilde{A}_n - A\right) \\ &= \sqrt{n}\left(\mathbb{\Lambda}_n - \lambda\right) f \\ &= \sqrt{n}\left(\frac{1}{n}\sum_{j=1}^{n} \max_{k=1\ldots d}(s_k^{(j)} w_k) - \mathbb{E}_{\mathbf{s}\sim\lambda}\left[\max_{k=1\ldots d}(s_k w_k)\right]\right), \end{aligned}$$

where $f \in \mathcal{F}$ and

$$\mathcal{F} := \{f_w(s) := \max_{k=1\ldots d}(s_k w_k) : \mathbf{w} \in \Delta_{d-1}\}.$$

By the classical central limit theorem, for a given $\mathbf{w}$, $\sqrt{n}\left(\tilde{A}_n(\mathbf{w}) - A(\mathbf{w})\right) \xrightarrow{d} \mathcal{N}(0, \sigma^2)$, with $\sigma^2 \leq (1 - A(\mathbf{w}))(A(\mathbf{w}) - \frac{1}{d^2})$, since the random variable $\max_{k=1\ldots d}(w_k s_k) \in [1/d^2, 1]$ is bounded and has finite variance.

Our claim is that $\mathbb{G}_n \rightsquigarrow \mathbb{G}$ where $\mathbb{G}$ is a zero-mean Gaussian process for establishing uniform convergence over $\mathbf{w}$. We will now show that the function class given by $\mathcal{F}$ is $\lambda-$Donsker. To show this, we will show that the bracketing integral given by

$$\mathcal{J}_{[]}(1, \mathcal{F}, L_2(\lambda)) = \int_0^1 \sqrt{\log N_{[]}(\epsilon, \mathcal{F} \cup 0, L_2(\lambda))}\, \mathrm{d}\epsilon \tag{17}$$

converges where the $L_2(\lambda)$ norm is defined as $\|f\|_{\lambda,2} = (\int f^2 d\lambda)^{1/2}$. A sufficient condition for convergence of (17) is to show that the logarithm of the bracketing number $N_{[]}$ grows at a rate slower than $O(\frac{1}{\epsilon^2})$. The function class $\mathcal{F}$ is indexed by $\mathbf{w} \in \Delta_{d-1}$ and is Lipschitz on $\mathbf{w}$. From Sen [2018, Lemma 2.14], the bracketing number of $\mathcal{F}$ is thus bounded above by the covering number of Δ_{d-1}, i.e.

$$N_{[]}(2\epsilon, \mathcal{F}, L_2(\lambda)) \leq N(\epsilon, \Delta_{d-1}, \|\cdot\|_2),$$

where the covering number of the unit simplex is asymptotically $O\left(\frac{1}{\epsilon^{d-1}}\right)$. The logarithm of the bracketing number then grows at a rate $\log N_{[]} \leq O\left((d-1)\log\frac{1}{\epsilon}\right) < O(\frac{1}{\epsilon^2})$. This proves that $\mathcal{F}$ is $\lambda-$Donsker and thus $\mathbb{G}_n \rightsquigarrow \mathbb{G}$. □

C SURVIVAL PROBABILITY ESTIMATION

One particularly useful task is estimating multi-dimensional survival probabilities rather than cumulative probabilities. More precisely, let $(\gamma_1, \cdots, \gamma_d) \in \mathbb{R}^d$ be a $d-$dimensional vector of thresholds, we are interested in calculating the following survival probability:

$$\mathbb{P}\left[M_n^{(1)} > \gamma_1, \cdots, M_n^{(d)} > \gamma_d\right] = \mathbb{P}\left[\bar{M}_n^{(1)} > \bar{\gamma}_1, \cdots, \bar{M}_n^{(d)} > \bar{\gamma}_d\right], \tag{18}$$

where $\bar{\gamma}_k = \frac{\gamma_k - b_n^{(k)}}{a_n^{(k)}}$, $k \in \{1, \cdots, d\}$.

To calculate this, we simply use a change-of-variable technique which we present in the following proposition. This approach is well known, and we only provide the proposition for completeness.

Proposition 1 (Survival Probability Computation)**.** *Let $G_k(x) := F_k^{-1}(1 - F_k(x))$ for $k \in \{1, \ldots, d\}$, then the random variables $G_k(\bar{M}_n^{(k)})$ and $\bar{M}_n^{(k)}$ have the same marginal CDF F_k, for $k \in \{1, \ldots, d\}$, and*

$$\mathbb{P}\left[\bar{M}_n^{(1)} > \bar{\gamma}_1, \cdots, \bar{M}_n^{(d)} > \bar{\gamma}_d\right] = \mathbb{P}\left[G_1(\bar{M}_n^{(1)}) < G_1(\bar{\gamma}_1), \cdots, G_d(\bar{M}_n^{(d)}) < G_d(\bar{\gamma}_d)\right]. \tag{19}$$

Proof. With the change-of-variable $G_k(x) := F_k^{-1}(1 - F_k(x))$ for $k \in \{1, \cdots, d\}$, it first follows that the random variables $G_k(\bar{M}_n^{(k)}) \sim F_k$:

$$\mathbb{P}(G_k(\bar{M}_n^{(k)}) \leq x) = \mathbb{P}(F_k^{-1}(1 - F_k(\bar{M}_n^{(k)})) \leq x) = \mathbb{P}(1 - F_k(\bar{M}_n^{(k)}) \leq F_k(x)) = F_k(x), \tag{20}$$

since $F_k(\bar{M}_n^{(k)})$ and $1 - F_k(\bar{M}_n^{(k)})$ follow the unit uniform distribution.

Moreover, the survival probability can be written as:

$$\begin{aligned}
\mathbb{P}\left[\bar{M}_n^{(1)} > \bar{\gamma}_1, \cdots, \bar{M}_n^{(d)} > \bar{\gamma}_d\right] &= \mathbb{P}\left[1 - F_1(\bar{M}_n^{(1)}) < 1 - F_1(\bar{\gamma}_1), \cdots, 1 - F_d(\bar{M}_n^{(d)}) < 1 - F_d(\bar{\gamma}_d)\right] \\
&= \mathbb{P}\left[G_1(\bar{M}_n^{(1)}) < G_1(\bar{\gamma}_1), \cdots, G_d(\bar{M}_n^{(d)}) < G_d(\bar{\gamma}_d)\right] \\
&= C\left(1 - F_1(\bar{\gamma}_1), \cdots, 1 - F_d(\bar{\gamma}_d)\right),
\end{aligned}$$

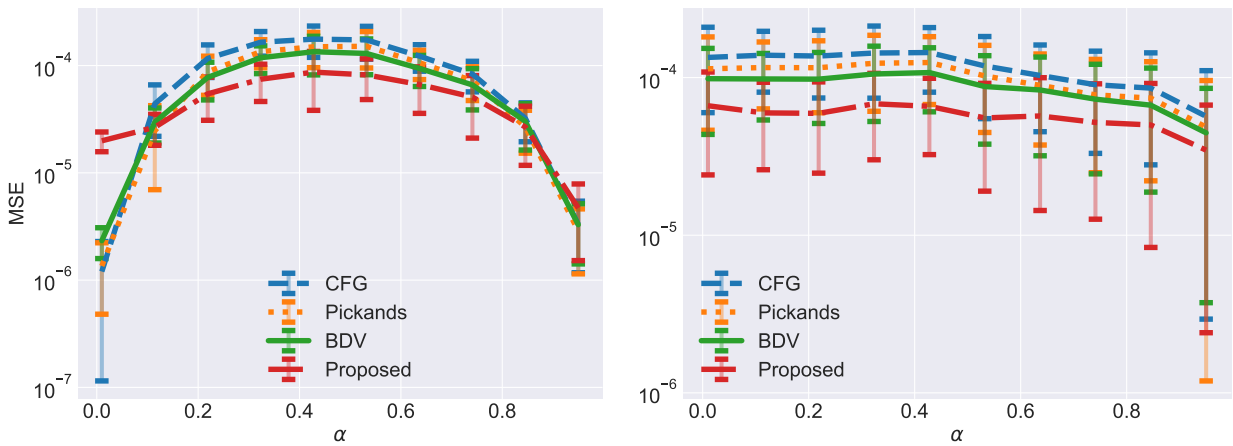
where C is the copula of $\left(G_1(\bar{M}_n^{(1)}), \cdots, G_d(\bar{M}_n^{(d)})\right)$. □

This proposition implies that the transformed variables $G_k(\bar{M}_n^{(k)})$ are samples from extreme value distributions. Then, we can fit Pickands dependence function to these transformed variables, and finally evaluate the corresponding extreme value copula on $(1 - F_1(\bar{\gamma}_1), \cdots, 1 - F_d(\bar{\gamma}_d))$. Details on how to estimate the survival probability in (18) are given in Algorithm 2.

D ADDITIONAL EXPERIMENTS AND FIGURES

D.1 24 WIDTH 3 DEPTH ARCHITECTURE

Here we repeat the experiments with a different architecture. All other hyperparameters are the same, the only difference is we increase the depth to 3 and use a width of 24 for each layer. Most of the results remain similar for the synthetic data but we see a change in the results for the real data, specifically, the Wind and Commodities data show a deterioration in performance. However, the variances are still high for the real experiments and not much can be said regarding the efficacy of any single method.



(a) A_{SL} MSE ($d = 2$)

(b) A_{ASL} MSE ($d = 2$)

Figure 7: Using 24 width 3 depth architecture: MSE of survival probabilities for $d = 2$ with 100 samples for A_{SL} (7a) and A_{ASL} (7b). Thresholds are above the 75th percentile.

	d	Train/Test Length	PICKANDS	CFG	BDV	PROPOSED
Wind	10	day/week	$4.48(18.6)_{\times 10^{-4}}$	*4.15(15.1)*$_{\times 10^{-4}}$	$\mathbf{4.10(16.3)_{\times 10^{-4}}}$	$4.80(20.6)_{\times 10^{-4}}$
Ozone	4	day/week	$3.06(4.66)_{\times 10^{-2}}$	$3.86(6.10)_{\times 10^{-2}}$	*2.86(4.46)*$_{\times 10^{-2}}$	$\mathbf{2.82(4.38)_{\times 10^{-2}}}$
Commodities	10	week/month	$4.34(5.82)_{\times 10^{-3}}$	$4.33(5.71)_{\times 10^{-3}}$	$\mathbf{1.60(1.96)_{\times 10^{-3}}}$	*2.20(3.41)*$_{\times 10^{-3}}$
S&P 500	418	week/month	*3.02(21.2)*$_{\times 10^{-3}}$	*3.02(21.1)*$_{\times 10^{-3}}$	$6.28(35.2)_{\times 10^{-3}}$	$\mathbf{2.41(22.2)_{\times 10^{-3}}}$
Crypto	100	week/month	$1.06(2.85)_{\times 10^{-2}}$	*1.05(4.86)*$_{\times 10^{-2}}$	$1.34(3.44)_{\times 10^{-2}}$	$\mathbf{8.42(26.1)_{\times 10^{-3}}}$

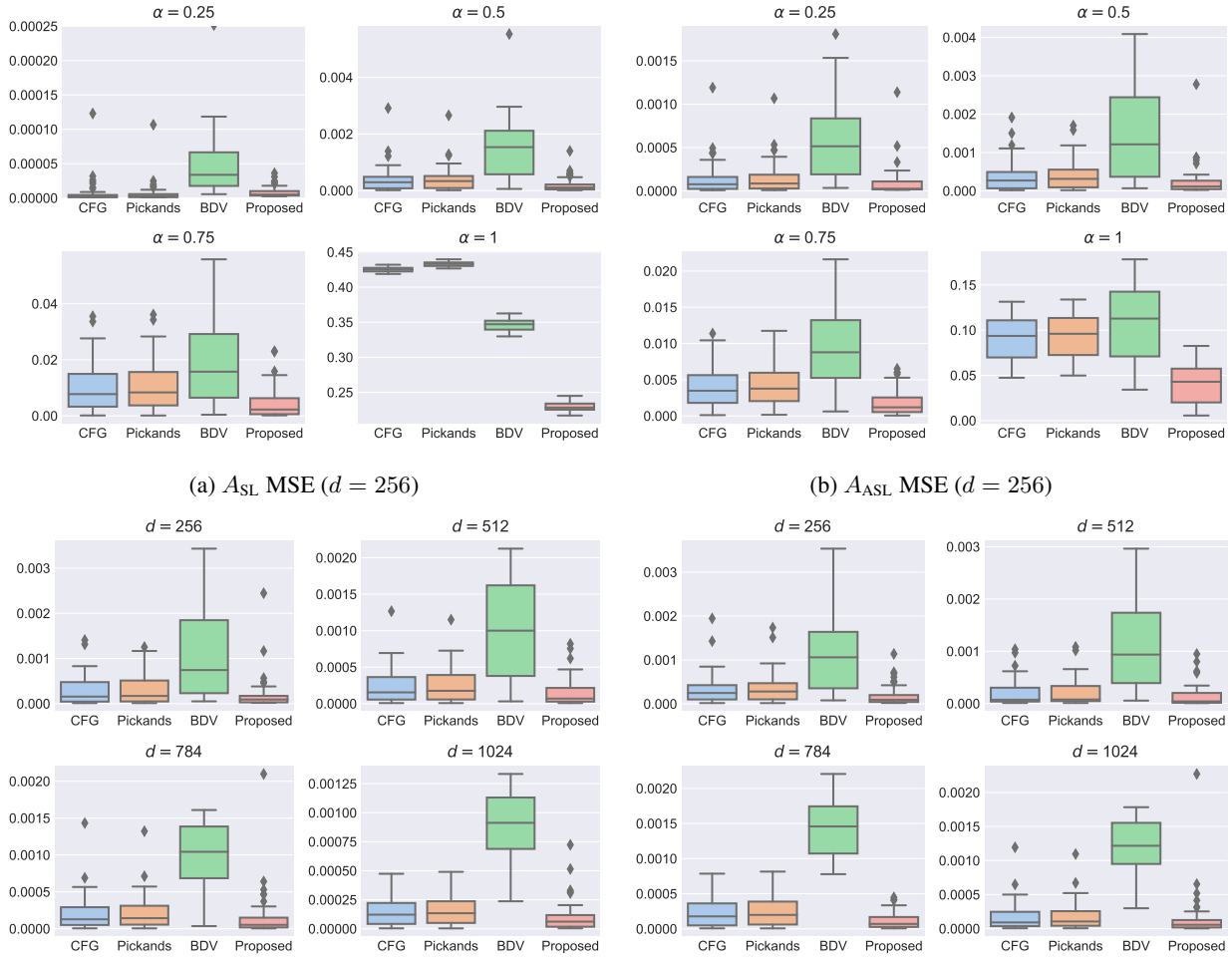
Table 3: MSE of different estimators in estimating maxima over two time scales for 24 width 3 depth architecture. Best and second best performances are marked in **bold** and *italic* respectively.

D.2 64 WIDTH 4 DEPTH ARCHITECTURE

Here we repeat the experiments with a different architecture. All other hyperparameters are the same, the only difference is we increase the depth to 4 and use a width of 64 for each layer. Most of the results remain similar for the synthetic data but we see a change in the results for the real data, specifically, the Wind and Commodities data show a deterioration in performance. However, the variances are still high for the real experiments and not much can be said regarding the efficacy of any single method.

D.3 ESTIMATION COMPARISON

We finally add a few figures comparing the learned dependence functions between different architectures. We additionally provide a table comparing the results for different architectures on the real data experiments in Table 5.



(a) A_{SL} MSE ($d = 256$)

(b) A_{ASL} MSE ($d = 256$)

(c) A_{SL} MSE ($\alpha = 0.5$)

(d) A_{ASL} MSE ($\alpha = 0.5$)

Figure 8: Using 24 width 3 depth architecture: Comparison of $||\hat{A}(w) - A(w)||_2^2$ for different estimators $\hat{A}$ for different dependence $\alpha = \{0.25, 0.50, 0.75, 1.0\}$ with fixed $d = 256$ (8a, 8b) and for fixed $\alpha = 0.5$ with different $d = \{256, 512, 728, 1024\}$ (8c, 8d). The truth models considered are A_{SL} (8a, 8c) and A_{ASL} (8b, 8d). Results are over 50 runs with 100 training samples for each run.

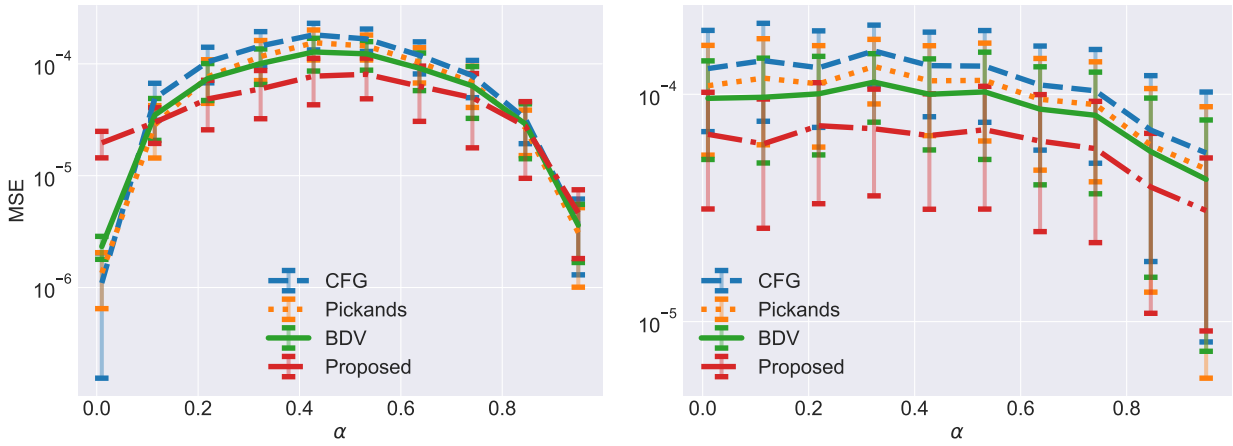
	d	Train/Test Length	PICKANDS	CFG	BDV	PROPOSED
Wind	10	day/week	$4.48(18.6)_{\times 10^{-4}}$	*4.15(15.1)*$_{\times 10^{-4}}$	$\mathbf{4.10(16.3)_{\times 10^{-4}}}$	$4.76(18.7)_{\times 10^{-4}}$
Ozone	4	day/week	$3.06(4.66)_{\times 10^{-2}}$	$3.86(6.10)_{\times 10^{-2}}$	*2.86(4.46)*$_{\times 10^{-2}}$	$\mathbf{2.73(4.25)_{\times 10^{-2}}}$
Commodities	10	week/month	$4.34(5.82)_{\times 10^{-3}}$	$4.33(5.71)_{\times 10^{-3}}$	$\mathbf{1.60(1.96)_{\times 10^{-3}}}$	*2.20(3.44)*$_{\times 10^{-3}}$
S&P 500	418	week/month	*3.02(21.2)*$_{\times 10^{-3}}$	*3.02(21.1)*$_{\times 10^{-3}}$	$6.28(35.2)_{\times 10^{-3}}$	$\mathbf{2.39(22.1)_{\times 10^{-3}}}$
Crypto	100	week/month	$1.06(2.85)_{\times 10^{-2}}$	*1.05(4.86)*$_{\times 10^{-2}}$	$1.34(3.44)_{\times 10^{-2}}$	$\mathbf{8.28(25.6)_{\times 10^{-3}}}$

Table 4: MSE of different estimators in estimating maxima over two time scales for 64 width 4 depth architecture. Best and second best performances are marked in **bold** and *italic* respectively.

E DATA DESCRIPTION

Synthetic Data

For the synthetic data experiments we consider samples of 100 points from each respective distribution. We use the full dataset for the batch size during training. We additionally sample 1000 points from the simplex for each data point during



(a) A_{SL} MSE ($d = 2$)

(b) A_{ASL} MSE ($d = 2$)

Figure 9: Using 64 width 4 depth architecture: (9a, 9b) MSE of survival probabilities for $d = 2$ with 100 samples for A_{SL} (9a) and A_{ASL} (9b). Thresholds are above the 75th percentile.

	512width $\times$ 1depth	24width $\times$ 3depth	64width $\times$ 4depth
Wind	$4.37(17.5)_{\times 10^{-4}}$	$4.80(20.6)_{\times 10^{-4}}$	$4.76(18.7)_{\times 10^{-4}}$
Ozone	$2.73(4.25)_{\times 10^{-2}}$	$2.82(4.38)_{\times 10^{-2}}$	$2.73(4.25)_{\times 10^{-2}}$
Commodities	$1.56(2.21)_{\times 10^{-3}}$	$2.20(3.41)_{\times 10^{-3}}$	$2.20(3.44)_{\times 10^{-3}}$
S & P	$2.41(22.2)_{\times 10^{-3}}$	$2.41(22.2)_{\times 10^{-3}}$	$2.39(22.1)_{\times 10^{-3}}$
Crypto	$8.57(26.4)_{\times 10^{-3}}$	$8.42(26.1)_{\times 10^{-3}}$	$8.28(25.6)_{\times 10^{-3}}$

Table 5: Comparison of 3 different architectures on the real data experiments.

training.

Ozone Data

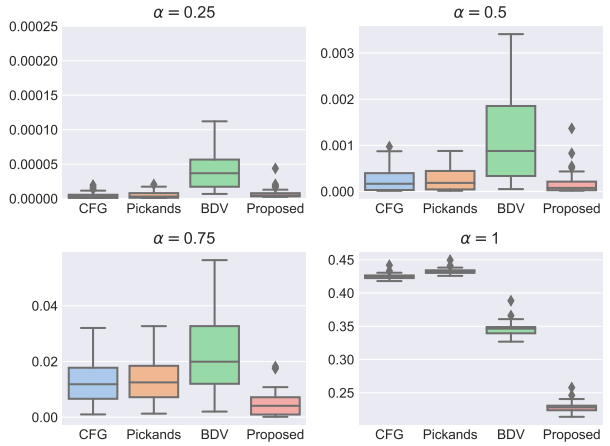
We consider ozone levels measured at 4 different stations in Sequoia National Park from data that can be downloaded from the National Park Service website [1]. The 4 stations are located at Ash Mountain, Lower Kaweah, Grant Grove and Lookout Point. We train the different models on daily maxima of ozone levels at the 4 different stations for the period from January 1984 to December 1996. To reduce the effect of seasonality, we do not train over the whole period, but we train different models on a single month (training month, e.g. June of each year) and compute accuracy on the consecutive month (validation month e.g. July of the same year). We additionally only look at summer months due to the increase of extreme events during that time. The accuracy is averaged with the specific validation month of each year over the whole period. We train on daily maxima and test on weekly maxima. For the experiments, we consider the following pair of (training/test) months: (June/July), (July/August), and (August/September).

California Wind Data

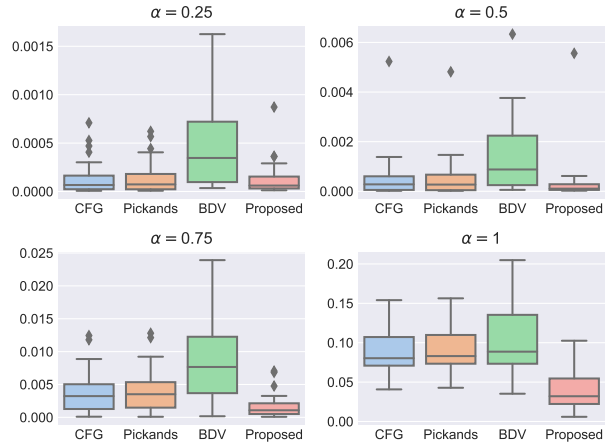
We are interested in modeling the extremal relationship of wind gusts between different locations in California during the summer months. We consider 10 locations in California illustrated in 17. We obtained the data from the Remote Automated Weather Station (RAWS) archive available at the online repository[2]. The RAWS data are collected from various time intervals from December 1989 to December 2020. We consider only the time points that occur in the intersection of all the data collected and where all values are valid (i.e. not NaNs or missing) for the summer months. Similarly to the ozone data, and in an effort to reduce seasonality, we consider the daily max wind gust for the different locations for a single month over

[1] `https://ard-request.air-resource.com/data.aspx`

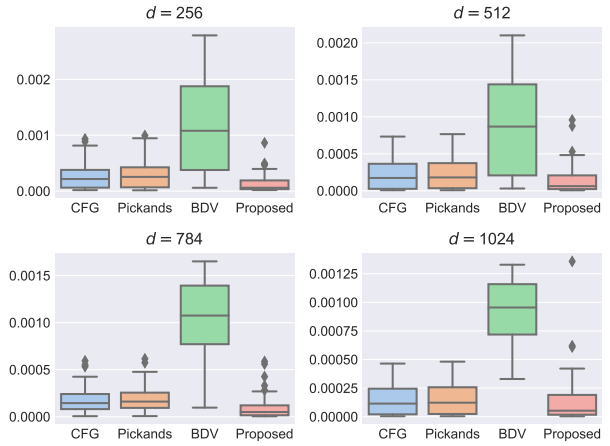
[2] `https://raws.dri.edu/index.html`



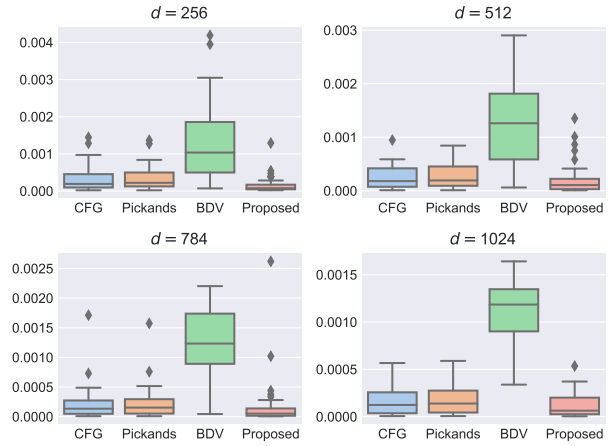
(a) A_{SL} MSE ($d = 256$)



(b) A_{ASL} MSE ($d = 256$)

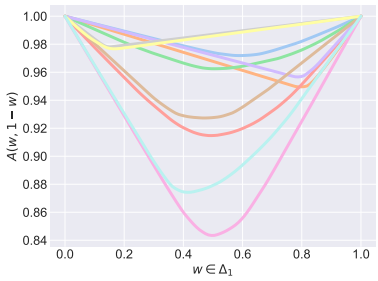


(c) A_{SL} MSE ($\alpha = 0.5$)

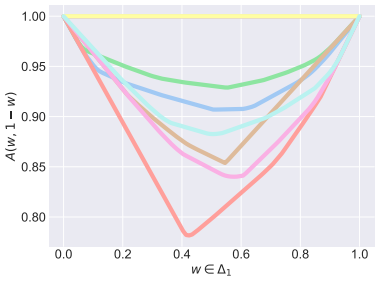


(d) A_{ASL} MSE ($\alpha = 0.5$)

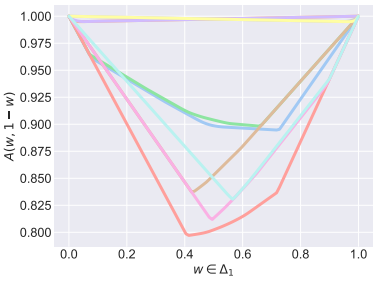
Figure 10: Using 64 width 4 depth architecture: Comparison of $||\hat{A}(w) - A(w)||_2^2$ for different estimators $\hat{A}$ for different dependence $\alpha = \{0.25, 0.50, 0.75, 1.0\}$ with fixed $d = 256$ (10a, 10b) and for fixed $\alpha = 0.5$ with different $d = \{256, 512, 728, 1024\}$ (10c, 10d) for A_{SL} (10a, 10c) and A_{ASL} (10b, 10d). Results are over 50 runs with 100 training samples for each run.



(a) 512 width 1 depth 2d Margins



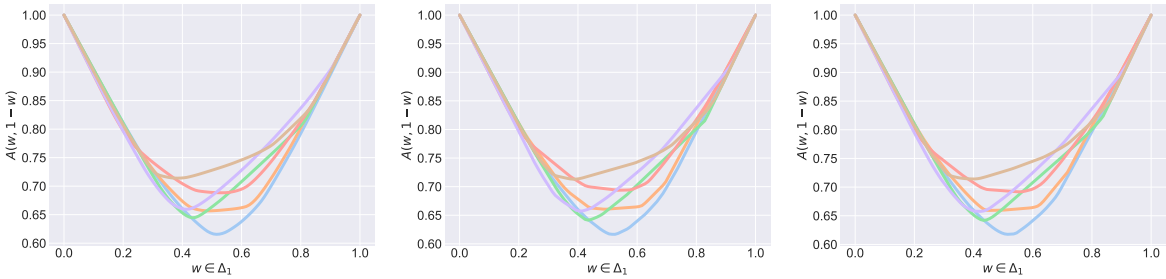
(b) 24 width 3 depth 2d Margins



(c) 64 width 4 depth 2d Margins

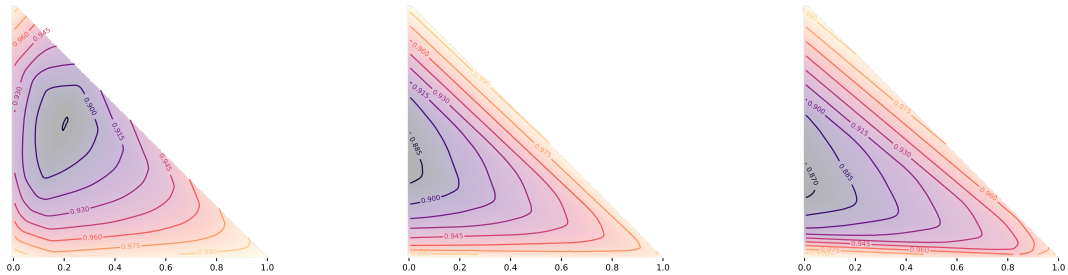
Figure 11: Margin comparison for winds dataset.

all the years the data were collected. To evaluate the proposed method, we train and test on data from consecutive months and repeat for multiple sets of months in our dataset. Additionally, we train on daily max and test on monthly max using the following data splitting scheme (training/validation months): (June/July), (July/August), and (August/September).



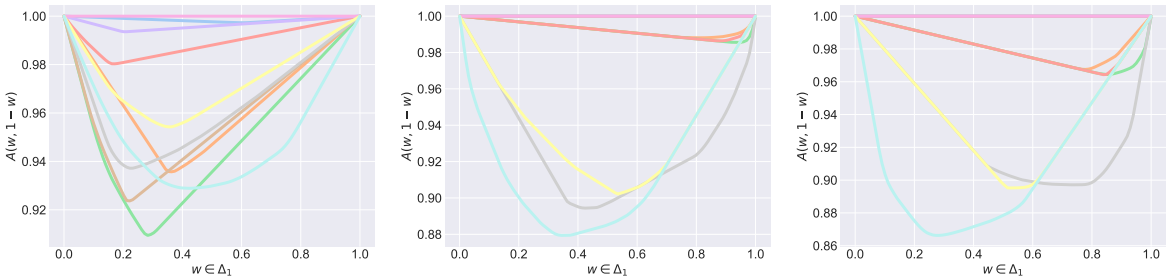
(a) 512 width 1 depth 2d Margins (b) 24 width 3 depth 2d Margins (c) 64 width 4 depth 2d Margins

Figure 12: Margin comparison for ozone dataset.



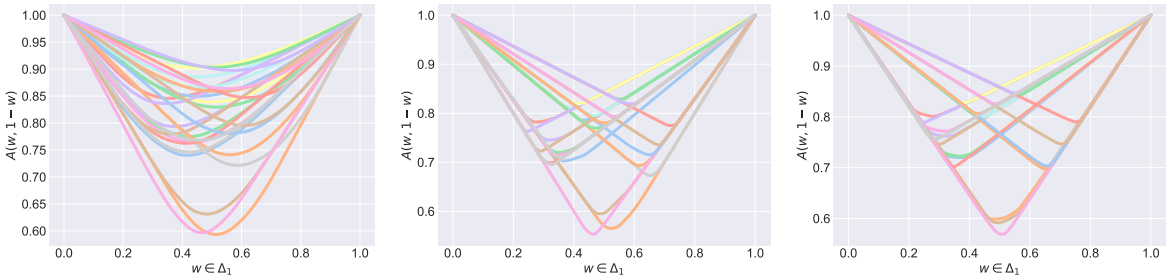
(a) 512 width 1 depth 3d Margins (b) 24 width 3 depth 3d Margins (c) 64 width 4 depth 3d Margins

Figure 13: Margin comparison for commodities dataset.



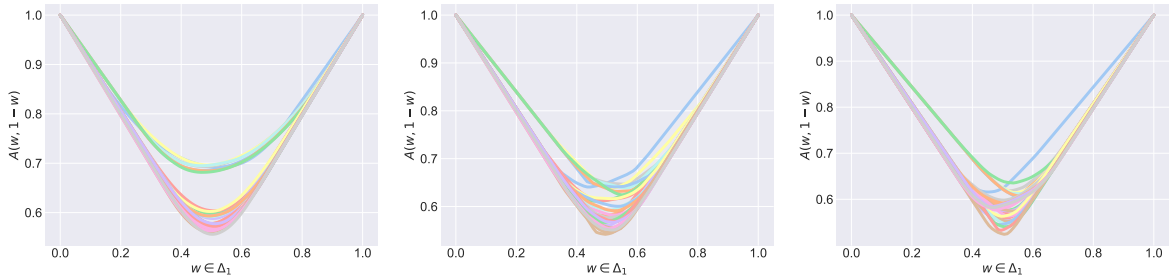
(a) 512 width 1 depth 2d Margins (b) 24 width 3 depth 2d Margins (c) 64 width 4 depth 2d Margins

Figure 14: Margin comparison for commodities dataset.



(a) 512 width 1 depth 2d Margins (b) 24 width 3 depth 2d Margins (c) 64 width 4 depth 2d Margins

Figure 15: Margin comparison for S&P 500 dataset.



(a) 512 width 1 depth 2d Margins (b) 24 width 3 depth 2d Margins (c) 64 width 4 depth 2d Margins

Figure 16: Margin comparison of different architectures for Cryptocurrencies dataset.

Commodities Data

We consider the extreme dependency between different commodities such as Coffee, Copper, Corn, Crude Oil, Gold, Heating Oil, Natural Gas, Platinum, Silver and Wheat. We collect data of daily prices of the different commodities from January 2015 to December 2020 as published in [3]. For training, we consider weekly max drawdown over a year. We validate the performance by evaluating accuracy of monthly max drawdown over next three years. We consider the following pairs of ([training years],[validation years]): ([2015], [2016, 2017, 2018]), ([2016], [2017, 2018, 2019]), ([2017], [2018, 2019, 2020]).

S&P 500 Data

We obtain historical data from `https://www.alphavantage.co`[4]. We choose the components of the S&P 500 with sufficient history (resulting in 418 stocks). For training, we consider weekly max drawdown over a year. We validate the performance by evaluating accuracy of monthly max drawdown over next three years. We consider the following pairs of ([training years],[validation years]): ([2015], [2016, 2017, 2018]), ([2016], [2017, 2018, 2019]), ([2017], [2018, 2019, 2020]). For the full list of stocks, see the `sp_names.txt` file in the supplementary materials.

Cryptocurrencies Data

We obtain historical data from `https://coinmarketcap.com`[5] for 100 coins with the longest history. For training, we consider weekly max drawdown over a year. We validate the performance by evaluating accuracy of monthly max drawdown over next three years. We consider the following pairs of ([training years],[validation years]): ([2015], [2016, 2017, 2018]), ([2016], [2017, 2018, 2019]), ([2017], [2018, 2019, 2020]). For the full list of coins, see the `crypto_names.txt` file in the supplementary materials.

F PICKANDS, CFG AND BDV ESTIMATORS

Pickands Estimator

The Pickands estimator Pickands [1981] is built following the transformations (6) and (7) in the paper. The estimator is obtained by exactly maximizing the likelihood (Equation (8) in the paper) resulting in the following non-parametric estimate:

$$\widehat{A}_{\text{Pickands}}(\mathbf{w}) = \left(\frac{1}{B} \sum_{i=1}^{B} Z_{w,i} \right)^{-1}. \tag{21}$$

[3] `https://www.investing.com/commodities/`

[4] Alpha Vantage allows academic use as long as the website is cited.

[5] Coin Market Cap allows academic use as long as the website is cited (see FAQ page).

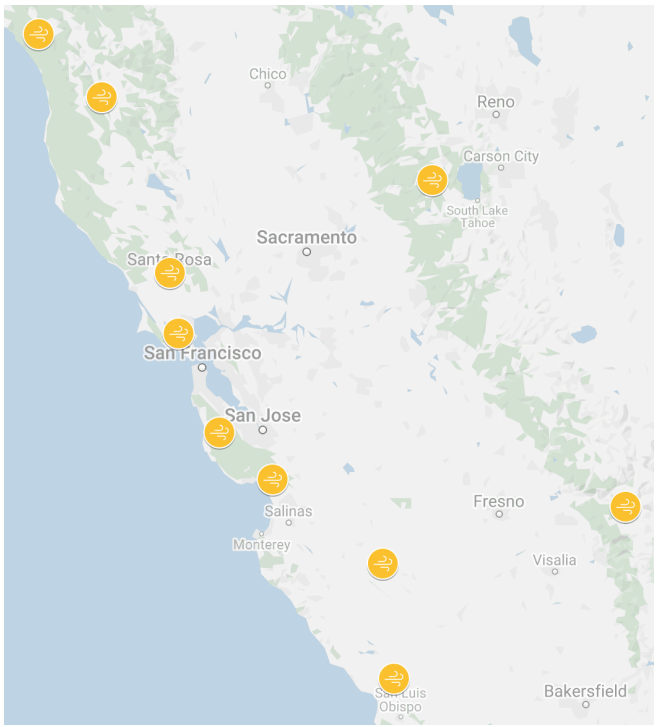


Figure 17: Locations (yellow circles) of weather stations sampled for the wind speed experiments. Figure generated via Google Maps.

CFG Estimator

The CFG estimator Capéraà et al. [1997] is constructed following the observation:

$$\mathbb{E} \log Z_w = -\log A(\mathbf{w}) - \gamma,$$

where $\gamma = -\int_0^\infty \log x e^{-x} dx$ denotes the Euler's contant. The CFG estimator is thus given by:

$$\widehat{A}_{\text{CFG}}(\mathbf{w}) = \exp\left[-\gamma - \frac{1}{B}\sum_{i=1}^{B} \log Z_{w,i}\right]. \tag{22}$$

In our main submission we use a similar estimator, with the correction term presented in Gudendorf and Segers [2011]:

$$\widehat{A}_{\text{CFG,C}}(\mathbf{w}) = \exp\left(\log \widehat{A}_{\text{CFG}}(\mathbf{w}) - \sum_{k=1}^{d} w_k \log\left(\widehat{A}_{\text{CFG}}(\mathbf{e}_k)\right)\right), \tag{23}$$

where $\mathbf{e}_k$ is the k-th canonical basis vector.

BDV Estimator

We propose an d-dimensional extension to the bivariate estimator described in Bücher et al. [2011]. We begin by defining the minimum distance estimator between the true CDF, $C(\mathbf{u})$ and the one estimated by the Pickands function $A(\mathbf{w})$.

$$\int_{[0,1]^d} \left[\log C(\mathbf{u}) - \sum_{k=1}^{d} \log u_k A\left(\frac{\log(\mathbf{u})}{\sum_k \log u_k}\right)\right]^2 d\mathbf{u} \tag{24}$$

$$= \int_{\Delta_{d-1}} \int_0^1 (\log C(y^{w_1}, \ldots, y^{w_d}) - \log(y) A(\mathbf{w}))^2 (-\log(y))^{d-1}\, dy\, d\mathbf{w}. \tag{25}$$

We have

$$\hat{C}(y^{w_1},\dots,y^{w_d}) = \frac{1}{B}\sum_{i=1}^{B}\mathbf{1}(F_1(\bar{M}_{n,i}^{(1)}) \le y^{w_1},\cdots,F_d(\bar{M}_{n,i}^{(d)}) \le y^{w_d}) \tag{26}$$

$$= \frac{1}{B}\sum_{i=1}^{B}\mathbf{1}(F_1(\bar{M}_{n,i}^{(1)})^{\frac{1}{w_1}} \le y,\cdots,F_d(\bar{M}_{n,i}^{(d)})^{\frac{1}{w_d}} \le y) \tag{27}$$

$$= \frac{1}{B}\sum_{i=1}^{B}\mathbf{1}(\max_{1\le k\le d} F_k(\bar{M}_{n,i}^{(k)})^{\frac{1}{w_k}} \le y) \tag{28}$$

$$= \frac{1}{B}\sum_{i=1}^{B}\mathbf{1}\left(\Gamma_{w,i} \le y\right), \tag{29}$$

where $\Gamma_{w,i} = \exp(-Z_{w,i})$. Now, if we reorder these so that $\Gamma_{w,1} \le \cdots \le \Gamma_{w,B}$, we have that

$$\hat{C}(y^{w_1},\dots,y^{w_d}) = \begin{cases} 0 \text{ if } y < \Gamma_{w,1}, \\ \dfrac{i}{B} \text{ if } \Gamma_{w,i} \le y < \Gamma_{w,i+1},\ i \in \{1,\dots,B-1\}, \\ 1 \text{ if } \Gamma_{w,B} \le y. \end{cases} \tag{30}$$

Because $\log \hat{C}(\cdots)$ is not defined if $y < \Gamma_{w,1}$, the following modified estimator is considered in Bücher et al. [2011].

$$\tilde{C}(y^{w_1},\dots,y^{w_d}) := \max\left\{C(y^{w_1},\dots,y^{w_d}), B^{-\gamma}\right\}, \tag{31}$$

where γ is any positive real greater or equal than $\frac{1}{2}$. For convenience, we choose $\gamma = 1$ so that:

$$\tilde{C}(y^{w_1},\dots,y^{w_d}) = \begin{cases} \dfrac{1}{B} \text{ if } y < \Gamma_{w,2}, \\ \dfrac{i}{B} \text{ if } \Gamma_{w,i} \le y < \Gamma_{w,i+1},\ i \in \{2,\dots,B-1\}, \\ 1 \text{ if } \Gamma_{w,B} \le y. \end{cases} \tag{32}$$

Finally, as in Bücher et al. [2011], for any positive weight function $h : (0,1) \to \mathbb{R}_0^+$, let $h^*(y) := h(y)(\log y)^2$,

$$B_h := \int_0^1 h^*(y)\,dy \quad \text{and} \quad g(x) := -B_h^{-1}\int_0^x \frac{h^*(y)}{\log y}\,dy. \tag{33}$$

Then, letting $\Gamma_{w,0} = 0$, $\Gamma_{w,B+1} = 1$, we define the BDV estimator $\widehat{A}_{\text{BDV},h}$ as follows

$$\widehat{A}_{\text{BDV},h}(\mathbf{w}) = B_h^{-1}\int_0^1 \frac{\log \tilde{C}(y^{w_1},\dots,y^{w_d})}{\log y} h^*(y)\,dy \tag{34}$$

$$= B_h^{-1}\sum_{i=0}^{B}\int_{\Gamma_{w,i}}^{\Gamma_{w,i+1}} \frac{\log \tilde{C}(y^{w_1},\dots,y^{w_d})}{\log y} h^*(y)\,dy \tag{35}$$

$$= -\log\frac{1}{n}g(\Gamma_{w,2}) - \sum_{i=2}^{n}\log\frac{i}{n}\left(g(\Gamma_{w,i+1}) - g(\Gamma_{w,i})\right) \tag{36}$$

$$= -\sum_{i=2}^{n}\log\frac{i-1}{n}g(\Gamma_{w,i}) + \sum_{i=2}^{n}\log\frac{i}{n}g(\Gamma_{w,i}) \tag{37}$$

$$= \sum_{i=2}^{n}\log\left(1+\frac{1}{i-1}\right)g(\Gamma_{w,i}) \tag{38}$$

In our main submission, we use a slightly modified estimator, which proved to have superior performance in our experiments. Recall that, if A is a Pickand's dependence function, we have $\max(\mathbf{w}) \le A(\mathbf{w}) \le 1$, which implies the true copula verifies:

$$\max(\mathbf{w}) \le \frac{\log C(y^{w_1},\dots,y^{w_d})}{\log y} = A(\mathbf{w}) \le 1. \tag{39}$$

Accordingly, we let

$$\mathrm{clamp}_{a,b}(x) := \begin{cases} a & \text{if } x \leq a, \\ x & \text{if } a < x < b, \\ b & \text{if } x \geq b, \end{cases} \tag{40}$$

and define

$$\breve{C}(y^{w_1}, \ldots, y^{w_d}) = \exp\left(\mathrm{clamp}_{\log y, \max(\mathbf{w}) \log y} \log \hat{C}(y^{w_1}, \ldots, y^{w_d})\right), \tag{41}$$

and

$$\widehat{A}_{\text{BDV,MM},h}(\mathbf{w}) = B_h^{-1} \int_0^1 \frac{\log \breve{C}(y^{w_1}, \ldots, y^{w_d})}{\log y} h^*(y)\, dy \tag{42}$$

$$= B_h^{-1} \sum_{i=0}^{B} \int_{\Gamma_{w,i}}^{\Gamma_{w,i+1}} \frac{\log \breve{C}(y^{w_1}, \ldots, y^{w_d})}{\log y} h^*(y)\, dy \tag{43}$$

Letting $\Gamma_{w,i}^{(\ell)} = \mathrm{clamp}_{\Gamma_{w,i},\Gamma_{w,i+1}}\left(\left(\frac{i}{n}\right)^{\frac{1}{\max(\mathbf{w})}}\right)$, $\Gamma_{w,i}^{(u)} = \mathrm{clamp}_{\Gamma_{w,i},\Gamma_{w,i+1}}\left(\frac{i}{n}\right)$ for $i \in \{0, \ldots, B\}$ and $\eta(x) = B_h^{-1} \int_0^x h^*(y)\, dy$, we have

$$\begin{aligned}
\widehat{A}_{\text{BDV,MM},h}(\mathbf{w}) &= B_h^{-1} \sum_{i=0}^{B} \int_{\Gamma_{w,i}}^{\Gamma_{w,i+1}} \frac{\log \breve{C}(y^{w_1}, \ldots, y^{w_d})}{\log y} h^*(y)\, dy \\
&= B_h^{-1} \sum_{i=0}^{B} \int_{\Gamma_{w,i}}^{\Gamma_{w,i}^{(\ell)}} \max(\mathbf{w}) h^*(y)\, dy + \int_{\Gamma_{w,i}^{(\ell)}}^{\Gamma_{w,i}^{(u)}} \log \frac{i}{n} \frac{h^*(y)}{\log y}\, dy + \int_{\Gamma_{w,i}^{(u)}}^{\Gamma_{w,i+1}} h^*(y)\, dy \\
&= \sum_{i=0}^{B} \max(\mathbf{w}) \left(\eta(\Gamma_{w,i}^{(\ell)}) - \eta(\Gamma_{w,i})\right) - \log \frac{i}{n} \left(g(\Gamma_{w,i}^{(u)}) - g(\Gamma_{w,i}^{(\ell)})\right) + \eta(\Gamma_{w,i+1}) \\
&- \eta(\Gamma_{w,i}^{(u)})
\end{aligned}$$

In our main submission, we use $\widehat{A}_{\text{BDV,MM},h}$ with $h(y) = \frac{1}{\log(y)}$.

G FURTHER DETAILS ON EXPERIMENTS

Architecture Details

For learning the Pickands dependence function, in all experiments in the manuscript we used 512 width and 1 depth dMNNs. Only the input layer was changed according to the input dimension. In order to force the weights to be positive, we use a weight clipping during training.

For the generative model experiments, we model p_z as a 128 d Gaussian random variable. The generator is a basic multi layer perceptron (MLP) with ReLU activations and batch norm. For all experiments, we use a width 256 and depth 2 MLP for the generator. The output is ensured to be positive through a final ReLU operation.

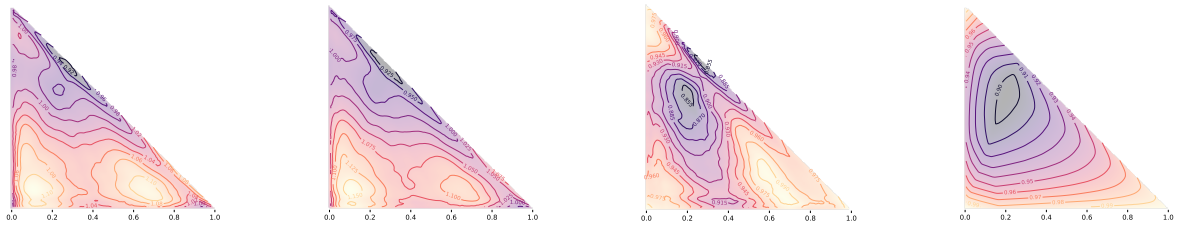
Hyperparameter Tuning

For learning the Pickands dependence experiments, we used the Adam Kingma and Ba [2014] optimizer for optimizing all parameters with learning rate 1×10^{-2} with a decay according to the `ReduceLROnPlateau` decay algorithm with a patience of 100 epochs. Each model was trained for 2000 epochs for the survival experiments and 4000 for the sampling experiments. For the sampling experiments, the generator was trained using Adam with learning rate 1×10^{-3}, $\beta_1 = 0.5$ and $\beta_2 = 0.99$ with exponential decay on the learning rate of 0.99998. Models for the generator were trained for 4000 epochs

Computational Resources

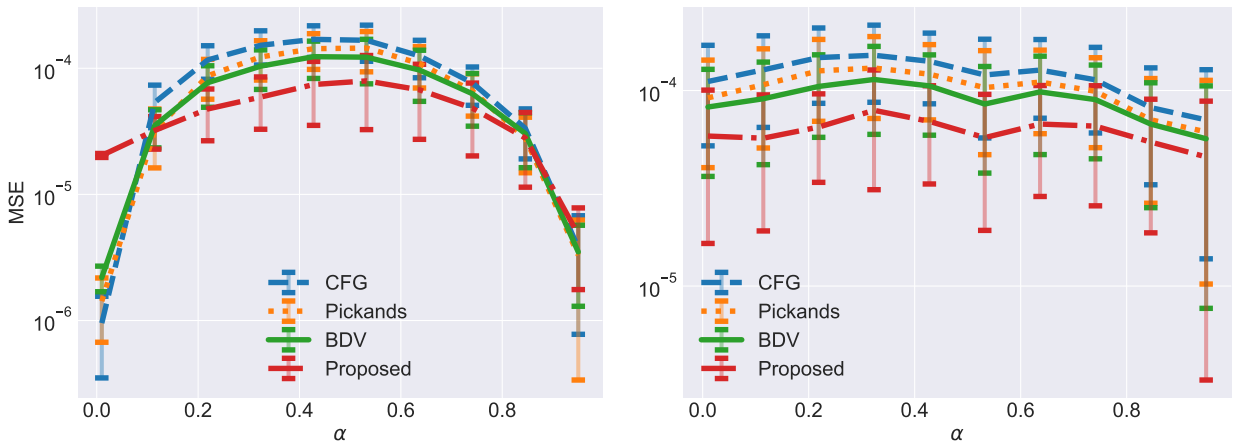
All experiments were run on an Nvidia RTX Titan GPU with an Intel Core i9-7900X CPU @ 3.30GHz and 64 GB of RAM.

H LARGER FIGURES



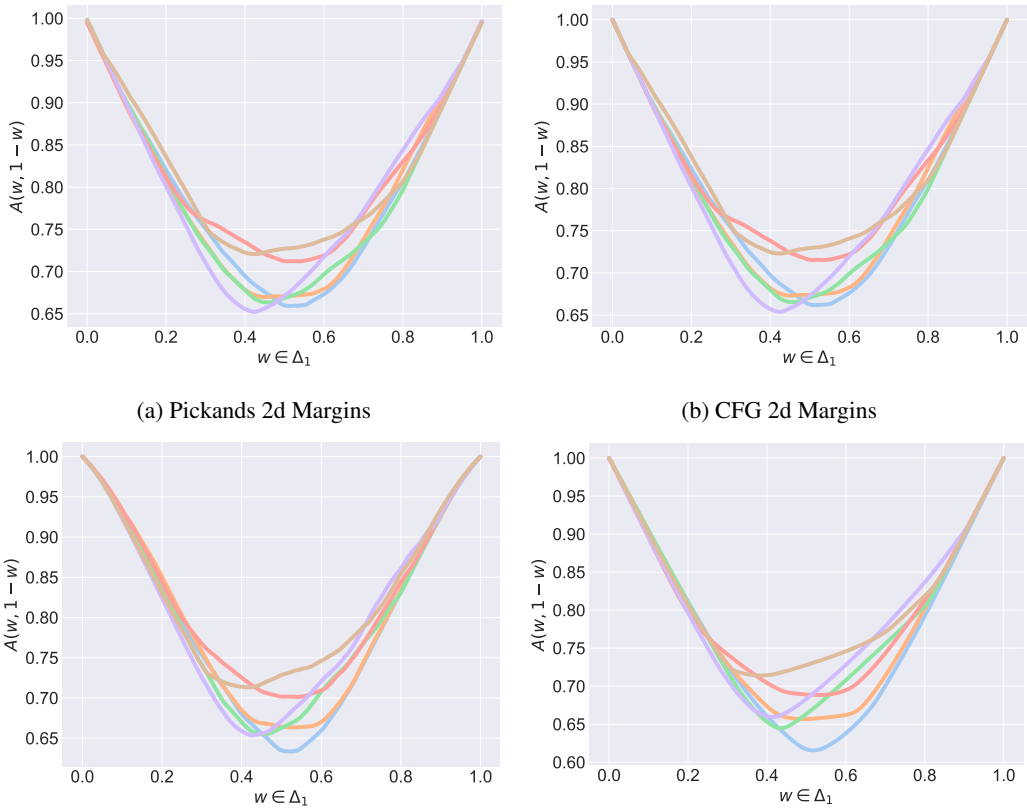
(a) Pickands 3d Margins (b) CFG 3d Margins (c) BDV 3d Margins (d) dMNN 3d Margins

Figure 18: (Larger figures from main text) Qualitative comparison of 3d margins from learned 10d MEV for the commodities dataset. The dMNN is the method that retains margins that are valid Pickands dependence functions as the others are non-convex and outside the required bounds. Contours plotted with solid line.



(a) A_{SL} MSE ($d = 2$) (b) A_{ASL} MSE ($d = 2$)

Figure 19: (Larger figures from main text) (3a, 3b) MSE of survival probabilities for $d = 2$ with 100 samples for A_{SL} (3a) and A_{ASL} (3b). Thresholds are above the 75th percentile.



(a) Pickands 2d Margins

(b) CFG 2d Margins

(c) BDV 2d Margins

(d) dMNN 2d Margins

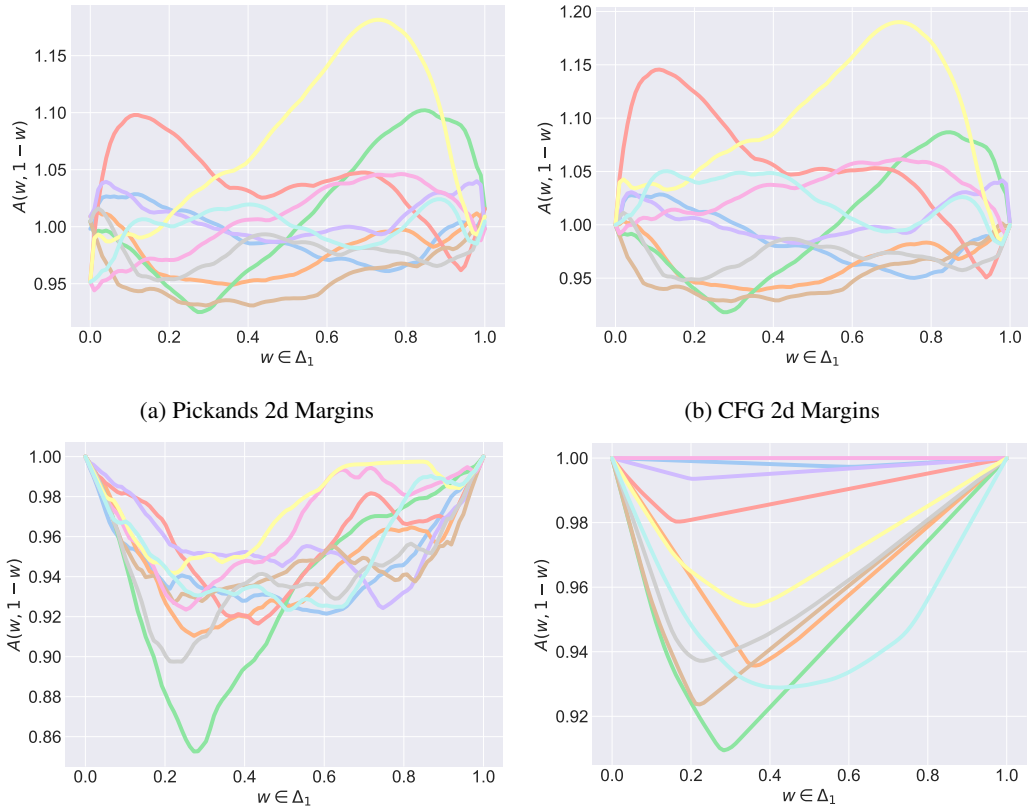
Figure 20: (Additional figure.) Qualitative comparison of 6 2d margins from learned 4d MEV for the Ozone dataset. The dMNN is the method that retains margins that are valid Pickands dependence functions as the others are non-convex and outside the required bounds.

I ALGORITHMS

Here we provide algorithms for the estimation and sampling presented in the main content.

Algorithm 2 Fitting the Pickands-dMNN to Data

1: **Input:** $\left\{\left(X_1^{(i)}, \ldots, X_d^{(i)}\right)\right\}_{i=1}^{N}$, $N = B \times n$ samples of i.i.d. random vectors where B is the number of blocks of data and n is the size of each block.
2: Take component-wise maxima over each block: $\left\{\left(M_1^{(n,b)}, \ldots, M_d^{(n,b)}\right)\right\}_{b=1}^{B}$ where $M_k^{(n,b)} = \max_{i=(b-1)n+1,\ldots,bn} X_k^{(i)}$, $(k, b) \in \{1, \ldots, d\} \times \{1, \ldots, B\}$.
3: Fit a GEV to each component-wise maxima $\{M_k^{(n,b)}\}_{b=1}^{B}$, obtain $\{\bar{M}_k^{(n,b)}\}_{b=1}^{B}$, then estimate marginals F_k for each $k \in \{1, \ldots, d\}$.
4: **Initialize** the parameters $\boldsymbol{\theta} \geq 0$ of the dMNN
Repeat:
5: Randomly sample a minibatch of training data $\{\bar{M}_k^{(n,b)}\}_{b \in \text{batch}}$ and uniformly sample $\mathbf{w} \in \Delta_{d-1}$.
6: Transform samples according to Equations (5) and (6) to obtain transformed samples $\{Z_{w,b}\}_{b \in \text{batch}}$.
7: Compute gradient $\nabla_{\boldsymbol{\theta}} \sum_{b \in \text{batch}} \mathcal{L}\left(Z_{w,b}; \boldsymbol{\theta}\right)$.
8: Update $\boldsymbol{\theta}$ with Adam [Kingma and Ba, 2014]
Until convergence
Output: $A_{\boldsymbol{\theta}}^{\star}(\mathbf{w})$.



(a) Pickands 2d Margins

(b) CFG 2d Margins

(c) BDV 2d Margins

(d) dMNN 2d Margins

Figure 21: (Additional figure.) Qualitative comparison of 10 2d margins from learned 10d MEV for the commodities. The dMNN is the method that retains margins that are valid Pickands dependence functions as the others are non-convex and outside the required bounds.

Algorithm 3 Estimating survival probabilities with the Pickands dependence function

1: **Input:** $\{\bar{M}_{n,b}^{(k)}\}_{b=1}^{B}$, thresholds: $(\gamma_1, \cdots, \gamma_d)$.

2: Train a model $A(\mathbf{w}; \theta)$ with the transformed variables $\{(G_1(\bar{M}_{n,b}^{(1)}, \ldots, G_d(\bar{M}_{n,b}^{(d)}))\}_{b=1}^{B}$ using Algorithm 1 and obtain $A(\mathbf{w}; \theta_*)$.

3: Evaluate the Pickands copula:

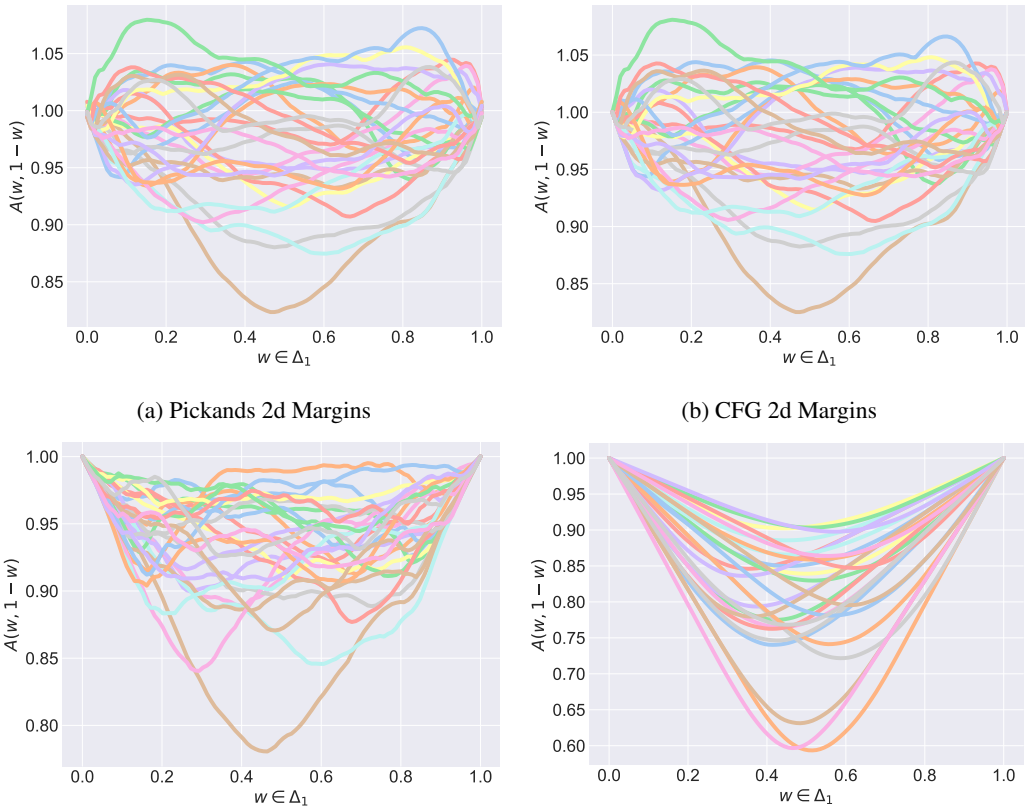
$$C\left(1 - F_1(\bar{\gamma}_1), \cdots, 1 - F_d(\bar{\gamma}_d)\right),$$

where C is calculated as in Equation (1) with $A = A(\mathbf{w}; \theta_*)$.

J SAMPLING EXAMPLES

K BACKGROUND ON EVT

The main idea behind EVT is to establish a form of the central limit theorem for the maxima of appropriately scaled random variables.



(a) Pickands 2d Margins

(b) CFG 2d Margins

(c) BDV 2d Margins

(d) dMNN 2d Margins

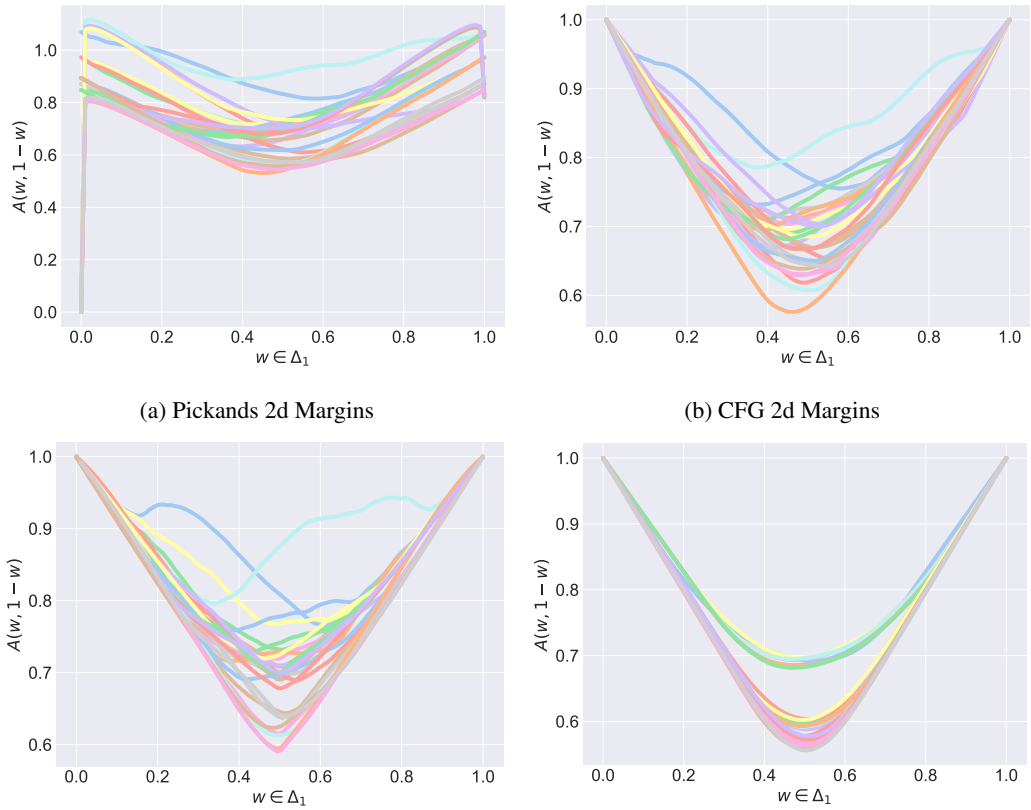
Figure 22: (Additional figure.) Qualitative comparison of 28 2d margins from learned 418d MEV for the S&P dataset. The dMNN is the method that retains margins that are valid Pickands dependence functions as the others are non-convex and outside the required bounds.

Algorithm 4 Training a Generator for a Pickands Copula

Input: $A(\mathbf{w})$, p_z, tolerance parameter ϵ
Initialize parameters ϕ of generator $G(\cdot;\phi)$
Sample $\{\mathbf{w}^{(j)}\}_{j=1}^{N_{\text{simplex}}}$ samples uniformly over Δ_{d-1}.
while $\sum_{j=1}^{N_{\text{simplex}}} \mathcal{L}(\mathbf{w}^{(j)};\phi) > \epsilon$ **do**
  Sample $\{\mathbf{w}^{(j)}\}_{j=1}^{N_{\text{simplex}}}$ samples uniformly over Δ_{d-1}.
  Sample $\{\mathbf{y}^{(i)}\}_{i=1}^{N_{\text{gen}}}$ where $\mathbf{y}^{(i)} = G(\mathbf{z}^{(i)};\phi), \mathbf{z}^{(i)} \sim p_z$ for $1 \leq i \leq N_{\text{gen}}$.
  Define $\eta(\mathbf{w},\mathbf{y}) = \max\{\mathbf{w} \odot \mathbf{y}\}$ with $\odot$ denoting the point-wise multiplication.
  Compute gradient w.r.t ϕ of $\sum_{j=1}^{N_{\text{simplex}}} \mathcal{L}(\mathbf{w}^{(j)};\phi)$ where:

$$\mathcal{L}(\mathbf{w}^{(j)};\phi) = \left\| A(\mathbf{w}^{(j)}) - \frac{1}{N_{\text{gen}}} \sum_{i=1}^{N_{\text{gen}}} \eta(\mathbf{w}^{(j)}, \mathbf{y}^{(i)}) \right\|_2^2 + \| \frac{1}{N_{\text{gen}}} \sum_{i=1}^{N_{\text{gen}}} \mathbf{y}^{(i)} - 1 \|_2^2$$

  Update ϕ using Adam Kingma and Ba [2014].
end while
Output: $G(.;\phi_*)$.



(a) Pickands 2d Margins

(b) CFG 2d Margins

(c) BDV 2d Margins

(d) dMNN 2d Margins

Figure 23: (Additional figure.) Qualitative comparison of 28 2d margins from learned 100d MEV for the Crypto dataset. The dMNN is the method that retains margins that are valid Pickands dependence functions as the others are non-convex and outside the required bounds.

Algorithm 5 Heuristic for Sampling From a Given Pickands Copula [Hofert et al., 2018, Algorithm 1]

Input: $A(\mathbf{w})$, $N_{\text{max}} > 1 \in \mathbb{N}$
Optimize a generator $G(\cdot; \phi)$ using Algorithm 4.
for $i \in \{1, \ldots, N_{\text{max}}\}$ **do**
  Generate $\mathbf{y}^{(i)}$ where $\mathbf{y}^{(i)} = G(\mathbf{z}^{(i)}; \phi_*)$, $\mathbf{z}^{(i)} \sim p_z$.
  Sample $\{\xi^{(i)}\}_1^{N_{\text{max}}}$ from the Poisson process by sampling $\epsilon_k \sim \text{Exp}(1)$ and $\xi^{(i)} = 1/\sum_{k=1}^{i} \epsilon_k$.
end for
Compute the component-wise maxima as: $M = \max_{1 \le i \le N_{\text{max}}} \{\xi^{(i)} \odot \mathbf{y}^{(i)}\}$.
Output: M.

K.1 MAIN DEFINITIONS AND THEOREMS

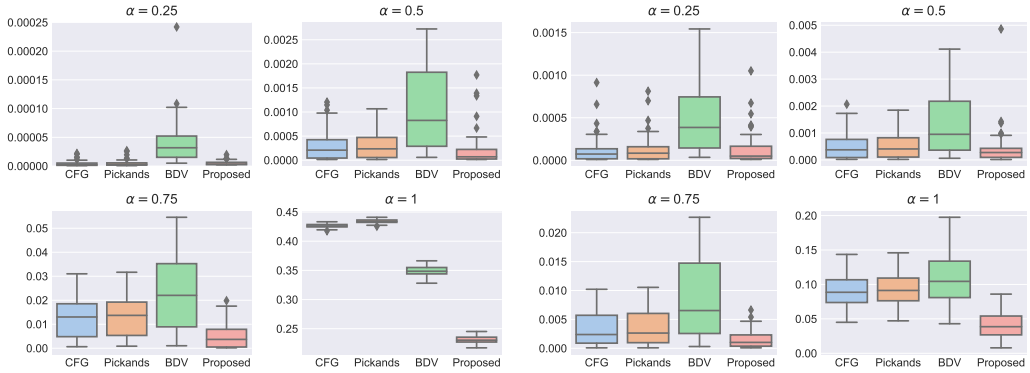
Theorem 3. *If C is a $d-$variate extreme value copula then there exists a tail dependence function $\ell : [0, \infty)^d \to [0, \infty)$ such that:*

$$C(u_1, \cdots, u_d) = e^{-\ell(-\log u_1, \cdots, -\log u_d)}, \tag{44}$$

where $(u_1, \cdots, u_d) \in (0, 1]^d$. Using the homogeneity property of ℓ, the extreme value copula C can be rewritten as:

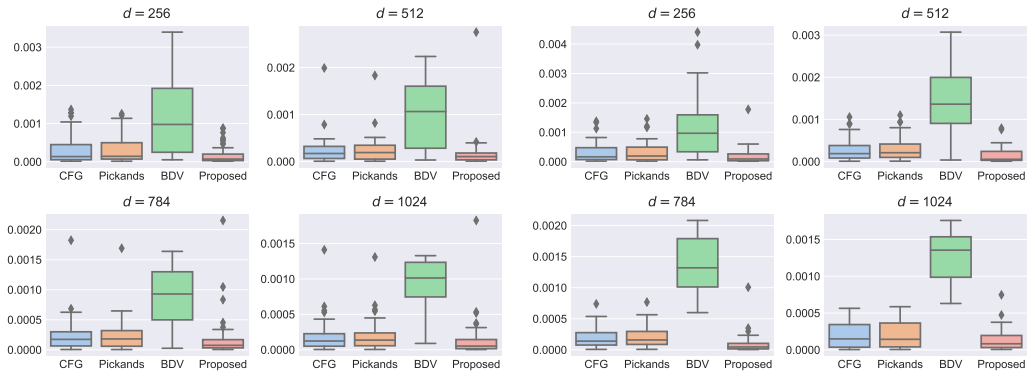
$$C(u_1, \cdots, u_d) = e^{\left(\sum_{k=1}^{d} \log u_k\right) A\left(\frac{\log u_1}{\sum_{k=1}^{d} \log u_k}, \cdots, \frac{\log u_d}{\sum_{k=1}^{d} \log u_k}\right)}, \tag{45}$$

where A is known as the Pickands dependence function, which can be thought of as the restriction of ℓ to the unit simplex $\Delta_{d-1} = \{\mathbf{w} = (w_1, \cdots, w_d) \in [0, \infty)^d : \sum_{k=1}^{d} w_k = 1\}$. The Pickands function A is known to be fully d-max-decreasing and satisfies:



(a) A_{SL} MSE ($d = 256$)

(b) A_{ASL} MSE ($d = 256$)



(c) A_{SL} MSE ($\alpha = 0.5$)

(d) A_{ASL} MSE ($\alpha = 0.5$)

Figure 24: (Larger figures from main text) Comparison of $||\hat{A}(\mathbf{w}) - A(\mathbf{w})||_2^2$ for different estimators $\hat{A}$ for different dependence $\alpha = \{0.25, 0.50, 0.75, 1.0\}$ and $d = 256$ (4a, 4b) and for fixed $\alpha = 0.5$ for $d = \{256, 512, 728, 1024\}$ (4c, 4d) for A_{SL} (4a, 4c) and A_{ASL} (4b, 4d). Results are over 50 runs with 100 training samples for each run.

$$\max_{1 \leq k \leq d} w_k \leq A(w_1, \cdots, w_d) \leq 1 \tag{46}$$

for all $\mathbf{w} = (w_1, \cdots, w_d) \in \Delta_{d-1}$.

Definition 3 (Tail dependence function)**.** *A function* $\ell : [0, \infty)^d \to [0, \infty)$ *is a tail dependence function if for all* $(x_1, \cdots, x_d) \in [0, \infty)^d$*, the following conditions are satisfied:*

- *(i)* ℓ *is fully d-max-decreasing and homogeneous of order* 1*, i.e.* $\ell(cx_1, \cdots, cx_d) = c \times \ell(x_1, \cdots, x_d)$*, for all* $c > 0$*.*
- *(ii)* $\max_{1 \leq k \leq d} x_k \leq \ell(x_1, \cdots, x_d) \leq \sum_{k=1}^{d} x_k$*.*

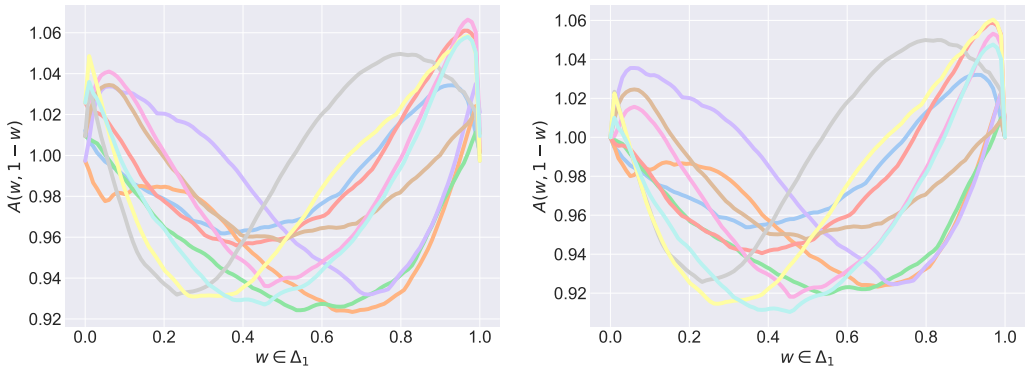
K.2 SPECTRAL DECOMPOSITION OF STATIONARY MAX-STABLE PROCESSES

Stationary max-stable processes can be intuitively interpreted as i.i.d. samples from infinite dimensional extreme value distributions (i.e. distributions over functions). A stationary max-stable process can be decomposed by the spectral representation defined in De Haan et al. [1984] which we recall in Proposition 2.

Proposition 2 (Spectral Representation of Max-Stable Processes [De Haan et al., 1984])**.** *Suppose that* $M(t)$ *has unit Fréchet margins and is stationary. Then,* $M(t)$ *can be written as:*

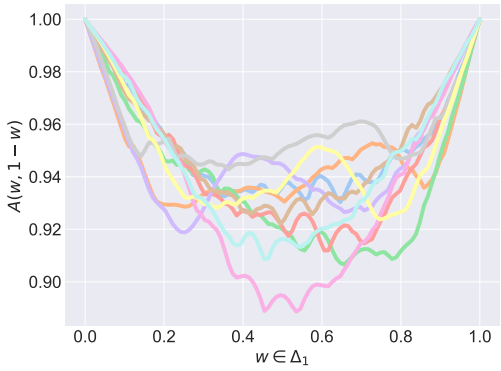
$$M(t) = \max_{i \geq 1} \xi_i Y_i^+(t), \quad t \in \mathcal{T}. \tag{47}$$

$\{Y_i(t)\}_{i \geq 1}$ *are i.i.d. copies of a continuous stochastic process* Y *defined on* $\mathcal{T}$ *such that* $\mathbb{E}[Y^+(t)] = 1$ *with* $Y^+(t) = \max\{0, Y(t)\}$ *and* ξ_i *is the* i*th realization of an independent Poisson point process on* $[0, \infty)$ *with intensity* $\xi^{-2} d\xi$*.*

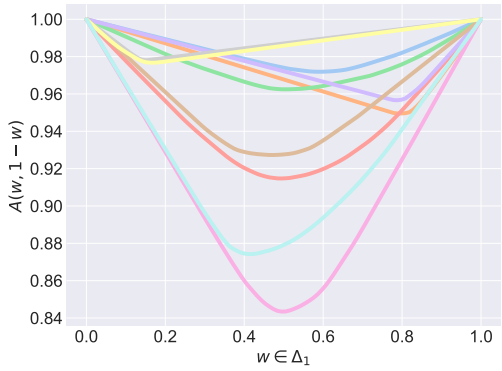


(a) Pickands 2d Margins

(b) CFG 2d Margins

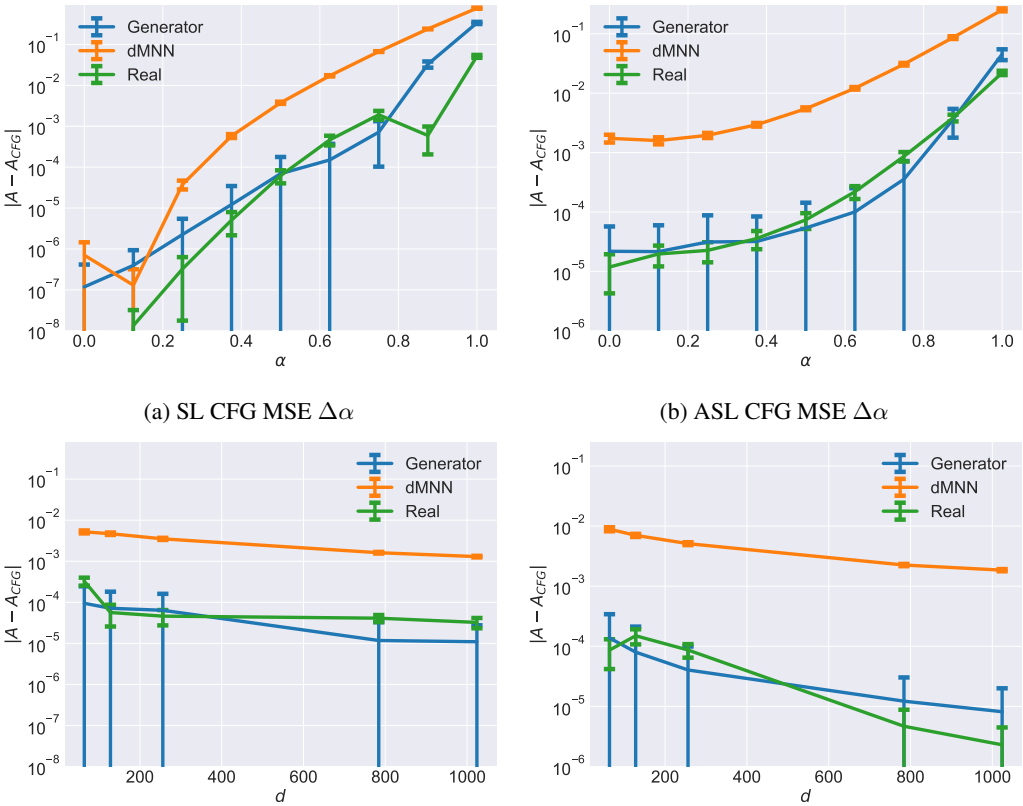


(c) BDV 2d Margins



(d) dMNN 2d Margins

Figure 25: (Larger figures from main text) Qualitative comparison of 10 out of 45 total 2d margins from learned 10d MEV for the California Winds dataset. The dMNN is the only method that retains margins that are valid Pickands dependence functions.



(a) SL CFG MSE $\Delta\alpha$

(b) ASL CFG MSE $\Delta\alpha$

(c) SL CFG MSE Δd

(d) ASL CFG MSE Δd

Figure 26: (Larger figures from main text) MSE of CFG estimate for 1000 samples and 1000 simplex points for $d = 225$ (6a, 6b) at various $\alpha \in (0, 1)$ and $\alpha = 0.5$ (6c, 6d) at $d = \{64, 128, 256, 784, 1024\}$ for A_{SL} (6a) and A_{ASL} (6b) for data sampled from generative model (blue), dMNN (orange), and exact sampled (green). Both models were trained with 1000 data points.