

High-Dimensional Portfolio Laws via Rank-Space Contraction

Daniel Grosshans^{1, 2} and Patrick Lucescu¹

¹Polykern, Stampfenbachstrasse 32, 8006 Zurich, Switzerland

²University of Basel, Missionsstrasse 62a, 4055 Basel, Switzerland

September 11, 2026

Abstract

Reliable and efficient evaluation of the distribution of a high-dimensional portfolio is essential for assessing and pricing risks. Yet, tractable dependence models often fail to capture the stronger dependence observed in extreme events, while more flexible approaches typically rely on simulation and become computationally prohibitive in rare tails. We develop an architecture that computes full portfolio distributions at polynomial cost with deterministic accuracy specified in advance. It separates frequently changing marginals, updated online, from a more slowly changing dependence layer prepared offline and reused. A rank-space representation combined with a first-tree Bernstein vine makes the computation simulation-free and scalable in dimension. Deterministic error guarantees hold uniformly across the distribution, including deep tails. The approach is measure agnostic: under risk-neutral inputs it delivers arbitrage-free pricing distributions for arbitrary baskets, while under physical inputs it provides distributional risk measures for efficient portfolio risk management.

Keywords. High-dimensional portfolio distributions; rank-space contraction; vine copulas; Bernstein copulas; deterministic error control; basket options; risk measures; comonotonic bounds.

Mathematics Subject Classification (2020). 91G60, 91G20, 91G10, 91G70, 62H05.

1 Introduction

Pricing claims on baskets and measuring risk in large portfolios both require a reliable joint law of many constituents at a common horizon. In high dimensions, this law must support frequent re-evaluation as inputs change, provide transparent error guarantees, and remain flexible enough to serve different applications built on the same underlying distribution. These requirements are particularly demanding in the tails, where economically important risks are concentrated and numerical reliability matters most.

Three properties are therefore central: flexibility in dependence, scalability, and reliable numerical error control. Simple Gaussian and correlation-based models are computationally convenient, but dependence summarized through global measures such as correlation is poorly suited to the asymmetric and concentrated co-movement observed in extreme markets (Longin and Solnik, 2001; Ang and Chen, 2002; Embrechts et al., 2002; Demarta and McNeil, 2005; McNeil et al., 2015). More flexible multivariate models can capture richer dependence, but in high dimensions they are typically evaluated by simulation. This shifts the computational bottleneck to sampling. As diversification pushes fixed-moneyness downside events deeper into the tail, maintaining a given relative precision requires an exponentially growing number of paths (Glasserman, 2004; Dembo and Zeitouni, 1998). Reliable numerical error control is therefore hardest to obtain precisely where it matters most. Conditional on a chosen model, one would ideally specify the desired accuracy *ex ante* together with the computational cost required to attain it. Existing approaches achieve such tractability either by imposing restrictive dependence structures or, under simulation, at sample sizes that can become prohibitive for rare tail events. Model-free methods provide robust ranges under limited information about dependence (Guo and Oblój, 2019; Bartl et al., 2020; Eckstein et al., 2021; Neufeld et al., 2023), but do not by themselves provide a precise point evaluation.

Our goal is to make flexible high-dimensional portfolio laws tractable for repeated evaluation. Sklar’s theorem (Sklar, 1959) provides the natural bridge by separating a joint law into its marginal distributions and its dependence structure. We use this decomposition computationally. Its importance is that the two inputs often move on very different clocks. Marginals can change frequently. In pricing, for example, liquid single-name option surfaces react continuously to new market information. Dependence, by contrast, is not directly observable, must typically be estimated from longer historical samples, and is therefore updated much less frequently. Our architecture exploits this asymmetry by moving the computationally expensive handling of dependence into an offline stage and retaining the marginal information required for highly efficient online recomputation. As marginals change, the portfolio law can therefore be updated without rebuilding the high-dimensional dependence structure from scratch.

The resulting architecture produces the full distribution of a high-dimensional portfolio at polynomial cost. Once the marginal laws and dependence model are specified, its numerical accuracy can be controlled in advance. In rank space, the marginals enter only through their quantile functions. Because the portfolio is additive, its law can be assembled along a one-dimensional aggregate grid rather than over a full high-dimensional state grid. To obtain an especially tractable implementation, we represent dependence by a depth-one vine with nonparametric Bernstein pair copulas. This choice gives small contraction states and permits the discretized portfolio law to be evaluated at polynomial cost. The rank-space architecture itself is more general: richer vine topologies and greater truncation depths can be accommodated by enlarging the contraction state. Depth one is therefore the computationally cheapest implementation rather than a fundamental restriction of the method. Tree-depth truncation is the one structural approximation in the dependence model. The remaining approximation errors are numerical, explicit, and controlled.

The output is a full portfolio distribution. With risk-neutral inputs, it produces basket values

and the entire strike curve at the chosen maturity, which can be recomputed as constituent option surfaces move. With physical inputs, it produces tail probabilities, quantiles, and risk measures. Because changes in individual marginals can be incorporated locally, the same architecture supports frequent repricing, scenario analysis, repeated risk evaluation, and distribution-sensitive portfolio optimization. The computational procedure is measure agnostic. Pricing, physical-measure risk analysis, and scenario applications are obtained by supplying the corresponding marginal laws and dependence model.

A central feature of the method is deterministic error control. Conditional on the supplied marginal and dependence inputs, the portfolio discretization gives an accuracy bound that is fixed before the calculation is run and does not deteriorate simply because the quantity of interest lies deep in the tail. The same absolute guarantee carries to prices, distribution functions, quantiles, and appropriate risk measures derived from the portfolio law. Tail completion and truncation are controlled separately, so numerical accuracy can be tightened transparently without re-estimating the dependence model.

Separating marginals from dependence also makes dependence risk explicit. Holding the marginals fixed, the same portfolio can be evaluated under a baseline dependence model and under alternative or deliberately conservative dependence assumptions. In risk management, this provides a direct way to report dependence stress scenarios and conservative benchmarks for aggregate losses. In pricing, the comonotonic coupling provides a dependence-robust upper benchmark for convex basket claims and, for positive weights, can be constructed from constituent option prices. This gives a natural conservative reference for applications such as margin requirements. The constructed pricing law is also internally free of static arbitrage at the modeled maturity. When the risk-neutral marginals are calibrated to constituent option prices, those options are repriced exactly, while put–call parity, monotonicity, convexity in strike, and the comonotonic bound hold by construction.

The resulting computational architecture remains practical at dimensions well beyond those used in the financial applications below. A controlled numerical benchmark demonstrates accurate full-law evaluation for a 100-constituent portfolio in under 0.1 seconds on a single GPU.

Contributions. The contribution of this paper is fourfold. First, we develop a general rank-space architecture for evaluating the full law of a high-dimensional portfolio and establish explicit polynomial-cost results for its depth-one Bernstein-vine implementation, with deterministic finite-grid and tail-error control once the law inputs are specified. A single contraction delivers the distribution function, its quantiles, and one-dimensional payoff values derived from the same law. Second, we turn the Sklar decomposition into an offline–online computational architecture that separates slowly changing dependence from frequently changing marginals and thereby enables efficient repeated evaluation. Third, the computational architecture is measure agnostic and supports pricing, physical-measure risk analysis, scenario evaluation, and parallel dependence benchmarking using the same computational core. Fourth, the construction preserves important financial structure. Under pricing inputs it is consistent with the supplied constituent marginals and static no-arbitrage restrictions, while under both pricing and risk inputs it provides conservative dependence benchmarks, including the comonotonic case.

Related literature. The paper connects several literatures that address different parts of this problem. Classical basket methods use moment matching, conditioning, comonotonicity, or Fourier techniques to approximate or bound the law of a sum (Levy, 1992; Ju, 2002; Deelstra et al., 2004; Dhaene et al., 2002b,a; Hobson et al., 2005; Carr and Madan, 1999; Hurd and Zhou, 2010; Caldana and Fusai, 2013; Caldana et al., 2016; Linders and Stassen, 2016). These methods obtain tractability

through approximations or through specific structural assumptions on the underlying model. Robust and model-free methods instead characterize no-arbitrage ranges under partial information about dependence (Tankov, 2011; Embrechts et al., 2013; Guo and Obłój, 2019; Bartl et al., 2020; Eckstein et al., 2021; Neufeld et al., 2023). They preserve dependence uncertainty explicitly, but return ranges rather than a single aggregate distribution under a chosen dependence specification. The risk-aggregation literature studies how dependence affects aggregate tail risk and the extremal bounds implied by incomplete dependence information (Bernard et al., 2017; Puccetti and Wang, 2015; Embrechts et al., 2015, 2016; McNeil et al., 2015). Our approach complements these strands by conditioning on a selected dependence model or scenario and computing the entire aggregate law efficiently, with deterministic numerical error attached to the resulting functionals.

On the computational side, simulation and deep-learning pricers address the high-dimensional bottleneck through sampling or learned approximations (Glasserman, 2004; Han et al., 2018; Becker et al., 2019). Deterministic and transform-based alternatives exploit regularity, sparsity, or low-rank structure in a specified transition law or pricing integrand (Gaß et al., 2018; Glau et al., 2020; Fang and Oosterlee, 2008; Pagès and Printems, 2003; Oseledets, 2011). More recent work pursues several complementary routes to high-dimensional multi-asset valuation. Bayer et al. (2023) combine optimized Fourier damping with adaptive sparse-grid quadrature, Junike and Stier (2025) develop multidimensional COS methods with convergence and error-control results, and Bayer et al. (2026) use randomized quasi-Monte Carlo in Fourier space with model-aware domain transformations. Low-rank alternatives instead compress the multidimensional objects themselves using tensor trains (Kastoryano and Pancotti, 2022; Arenstein and Kastoryano, 2026). These approaches seek exploitable structure in the high-dimensional pricing integral. Our construction places that structure directly in the representation of the joint law.

For dependence modeling, we build on established Bernstein and vine copula methods (Sancetta and Satchell, 2004; Cottin and Pfeifer, 2014; Scheffer and Weiß, 2017; Bedford and Cooke, 2001, 2002; Aas et al., 2009; Brechmann et al., 2012; Dißmann et al., 2013; Czado, 2019). These methods provide flexible multivariate dependence representations that are typically estimated independently of the subsequent portfolio calculation. Our contribution is to use the first-tree Bernstein representation itself as a contractible representation of the portfolio law. Marginal laws enter through arbitrary quantile functions, while the dependence layer supplies the finite-width contraction structure. The resulting calculation returns the full aggregate portfolio distribution and supports local marginal updates without reconstructing the high-dimensional dependence model.

Throughout the paper we often use pricing applications because option-implied marginals provide a particularly transparent example of rapidly changing inputs and allow sharp no-arbitrage results. The computational results apply equally to physical-measure and scenario applications when the corresponding marginals and dependence model are supplied. Pricing and risk applications may therefore use different dependence specifications without changing the computational architecture.

Outline. The remainder of the paper develops these claims in increasing levels of detail. Section 2 derives the rank-space representation and the computational separation of marginals and dependence. Section 3 constructs the finite-lattice portfolio law and establishes its error and consistency properties. Section 4 introduces the first-tree Bernstein-vine dependence layer, and Section 5 gives the exact tree contractions. Section 6 establishes polynomial complexity and compares the architecture with tensor grids and simulation. Section 7 treats tail completion and finite-level tail resolution, and Section 8 formalizes online marginal updates and cache maintenance. Section 9 discusses extensions beyond the first-tree construction, and Section 10 summarizes the construction of the dependence layer. Section 11 provides a controlled high-dimensional benchmark of numerical accuracy and runtime,

while Section 12 illustrates the resulting portfolio laws in two financial applications. Section 13 concludes.

2 The Sklar Bridge and the Rank-Space Portfolio Identity

We consider a family of d assets $(S_{r,T})_{r=1}^d$, where the joint law is governed by a copula C_t and each asset $S_{r,T}$ is endowed with a one-dimensional marginal law $F_{r,t}$. Throughout, t is the valuation date and T the portfolio-law horizon.

Given weights $(w_r)_{r=1}^d$, we are interested in the law of the basket B_T and the associated European claim values, where the basket is defined as

$$B_T = \sum_{r=1}^d w_r S_{r,T}. \quad (1)$$

Under mild regularizing conditions, Sklar's theorem (Sklar, 1959) allows us to express the joint probability density as

$$p_t(s) = c_t(F_{1,t}(s_1), \dots, F_{d,t}(s_d)) \prod_{r=1}^d f_{r,t}(s_r), \quad (2)$$

where c_t and $f_{r,t}$ denote the densities of C_t and $F_{r,t}$, respectively. The decomposition assigns each input a natural update frequency. The dependence model is typically estimated or calibrated from information that changes relatively slowly and can therefore be prepared *offline*. The marginal laws may change much more frequently, for example with option surfaces, statistical forecasts, or scenario inputs, and enter the calculation *online*. Figure 1 summarizes this split.

Using Equation (2), any integrable one-dimensional functional of the aggregate portfolio can be expressed as an integral over the d -dimensional rank cube. The density representation in (2) is useful when the joint law is absolutely continuous, but the rank-space identity itself does not require this assumption.

Theorem 1 (Rank-space portfolio identity). *Let C_t be a copula on $[0, 1]^d$ and $\Phi : \mathbb{R} \rightarrow \mathbb{R}$ a one-dimensional function such that $\Phi(B_T)$ is integrable. If $Q_{r,t}$ is the generalized quantile of $F_{r,t}$ defined as*

$$Q_{r,t}(u) := \inf\{x \in \mathbb{R} : F_{r,t}(x) \geq u\}, \quad 0 < u < 1, \quad (3)$$

then

$$\mathbb{E}_t[\Phi(B_T)] = \int_{[0,1]^d} \Phi\left(\sum_{r=1}^d w_r Q_{r,t}(u_r)\right) dC_t(u). \quad (4)$$

If C_t is absolutely continuous with density c_t , this reduces to

$$\mathbb{E}_t[\Phi(B_T)] = \int_{[0,1]^d} \Phi\left(\sum_{r=1}^d w_r Q_{r,t}(u_r)\right) c_t(u) du. \quad (5)$$

Proof. Let $U = (U_1, \dots, U_d)$ have uniform marginals and copula C_t . Under (3), the vector

$$S_T = (Q_{1,t}(U_1), \dots, Q_{d,t}(U_d))$$

has marginals $F_{r,t}$ and the joint law induced by C_t through the generalized-quantile construction. Hence,

$$\Phi(B_T) = \Phi\left(\sum_{r=1}^d w_r Q_{r,t}(U_r)\right),$$

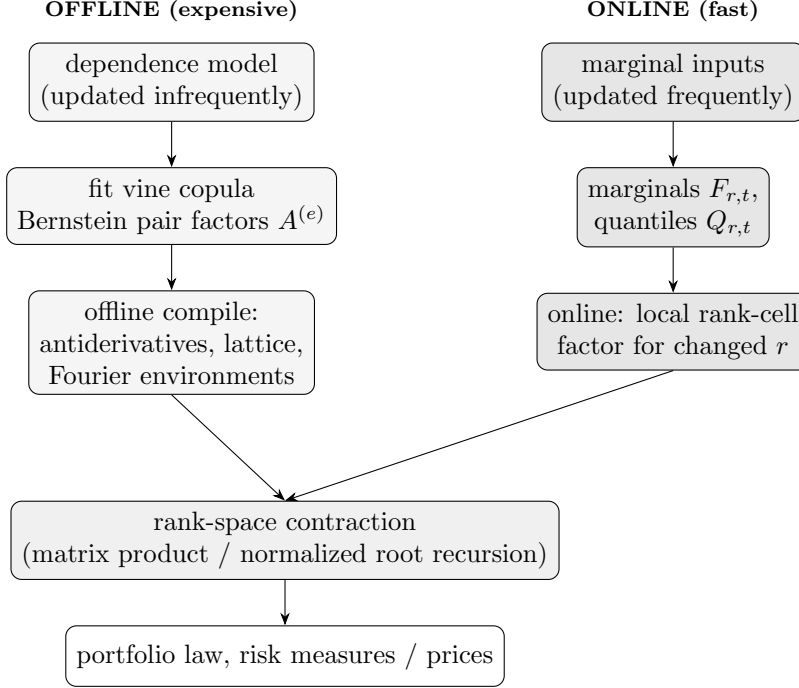


Figure 1: The offline/online split induced by Sklar’s theorem. The slowly varying dependence layer is compiled offline into contraction-ready objects; frequently updated one-dimensional marginals enter online through rank endpoints and are folded into the pre-compiled contraction. For the D-vine implementation, a baseline-relative single-marginal update recomputes one factor irrespective of d , at cost $O(R^2 L \log L)$ in the lattice length L , where $R = n_c + 1$.

and taking expectation with respect to the law C_t of U yields (4). If C_t has density c_t , then $dC_t(u) = c_t(u) du$, which gives (5). $\square$

Remark 1 (Atoms). If $F_{r,t}$ has an atom of mass a at s_0 , then $Q_{r,t}$ is constant at s_0 on a rank interval of length a , and each copula basis factor must be integrated over that full interval—the atom must not be attached independently in the native domain (Appendix A).

Remark 2 (Continuous marginals). If the marginals are continuous with densities, the density version of the identity is also obtained by substituting $u_r = F_{r,t}(s_r)$ in the usual Sklar density formula, since $du_r = f_{r,t}(s_r) ds_r$.

3 Finite-Lattice Approximation of the Portfolio Law

Theorem 1 separates the joint law into two distinct components: the one-dimensional marginals, which enter only through their generalized quantile maps, and the dependence structure, which acts on the latent ranks alone. This separation is exact, but it does not by itself make evaluation of the aggregate portfolio law cheap: the resulting expectation remains a d -dimensional integral over the unit cube, so the curse of dimensionality is still present.

We reduce this integral in two steps. First, each weighted constituent contribution is quantized on a common one-dimensional lattice, turning the aggregate portfolio into a finite discrete random variable. Second, we represent the copula density by a finite sum of products of univariate local factors. This separable structure turns joint cell probabilities into products of one-dimensional local

masses and the full portfolio law into a discrete convolution along a single aggregate axis. The same locality supports efficient online re-evaluation: the dependence layer is compiled once offline, while a change in one marginal requires recomputing only the corresponding local factor. Once the aggregate law has been obtained, CDFs, quantiles, risk measures, and, under pricing inputs, complete strike curves follow from the same output.

3.1 Marginal discretization

For each marginal $F_{r,t}$, we assume a bounded support $[\underline{s}_r, \bar{s}_r]$ ¹ The support of the weighted contribution of constituent r is then

$$[a_r, b_r] = [\min\{w_r \underline{s}_r, w_r \bar{s}_r\}, \max\{w_r \underline{s}_r, w_r \bar{s}_r\}]. \quad (6)$$

Using the auxiliary variables $Z_r = w_r Q_{r,t}(U_r) - a_r \in [0, W_r]$, with $(U_1, \dots, U_d)$ defined as in Theorem 1 and $W_r = b_r - a_r$, the aggregate portfolio can be written as

$$B_T := \kappa + \sum_{r=1}^d Z_r(U_r), \quad (7)$$

where $\kappa = \sum_{r=1}^d a_r$. The shift introduced by the auxiliary variables ensures we can accommodate negative weights without requiring copula reflection.

For a desired aggregate-axis spacing h , the number of grid points for each marginal domain $[0, W_r]$ is

$$M_r := \max \left\{ 1, \left\lceil \frac{W_r}{h} \right\rceil \right\}, \quad (8)$$

and the local cells and their representative centers are

$$I_{r,k} := [kh, (k+1)h), \quad z_{r,k} := \left(k + \frac{1}{2}\right)h, \quad k = 0, \dots, M_r - 1, \quad (9)$$

with the final cell closed at its right endpoint. While the last cell may extend to $M_r h > W_r$, mass outside $[0, W_r]$ is zero. Define the midpoint quantizer by

$$D_{r,h}(z) := \sum_{k=0}^{M_r-1} z_{r,k} \mathbf{1}\{z \in I_{r,k}\}, \quad z \in [0, W_r], \quad (10)$$

which maps every point in a cell to that cell's midpoint. The quantized basket value is then expressed as

$$B_{T,h} := \kappa + \sum_{r=1}^d D_{r,h}(Z_r(U_r)), \quad (11)$$

and can take possible values

$$b_n^{(h)} = \kappa + \left(n + \frac{d}{2}\right)h, \quad n = 0, \dots, M_\Sigma - 1, \quad M_\Sigma := 1 + \sum_{r=1}^d (M_r - 1). \quad (12)$$

¹Unbounded supports are handled by truncation or completion, with a separately controlled approximation discussed in Section 7.

3.2 Copula factorization and the exact pushforward

We define a finite factorization of the copula density as

$$c_t(u) = \sum_{\eta \in \mathcal{I}} \lambda_\eta \prod_{r=1}^d g_{r,\eta_r}(u_r), \quad \lambda_\eta \geq 0, \quad (13)$$

where the index set $\mathcal{I}$ can be arbitrarily chosen and each g_{r,η_r} is a one dimensional function mapping the unit interval to positive real numbers. Under this factorization and the marginal discretization introduced in Section 3.1, we can derive the exact law of the quantized aggregate portfolio.

Theorem 2 (Exact pushforward of the quantized portfolio law). *Let the copula density have the structure in Equation (13), and for each local factor define exact rank-cell mass as*

$$\mu_{r,\eta_r,k} := \int_0^1 g_{r,\eta_r}(u) \mathbf{1}\{Z_r(u) \in I_{r,k}\} du. \quad (14)$$

For each point $b_n^{(h)}$ in Equation (12), the probability assigned by $B_{T,h}$ is

$$\pi_n^{(h)} = \sum_{\eta \in \mathcal{I}} \lambda_\eta (\mu_{1,\eta_1} * \cdots * \mu_{d,\eta_d})_n, \quad (15)$$

where $*$ is ordinary discrete linear convolution over the cell index.

Proof. Partition each rank coordinate by the preimages $\{u : Z_r(u) \in I_{r,k}\}$. Fubini's theorem and (13) factor the probability of each cell-index vector $(k_1, \dots, k_d)$ into the product of the corresponding local integrals. Summing over all vectors with $\sum_r k_r = n$ gives the discrete convolution. $\square$

Thus replacing multiplication of one-dimensional factors by convolution of their local cell arrays is exact for the explicitly defined random variable $B_{T,h}$. No quadrature or grid approximation remains inside (15) when the local integrals (14) are evaluated exactly. The mass array $\{\pi_n^{(h)}\}$ fully determines the discrete portfolio law. In particular, its CDF and, under risk-neutral inputs, its undiscounted put prices are

$$F_{B_h}(K) = \sum_{b_n^{(h)} \leq K} \pi_n^{(h)}, \quad \Pi_h(K) = \sum_n (K - b_n^{(h)})^+ \pi_n^{(h)}. \quad (16)$$

The same mass array yields quantiles and other one-dimensional risk functionals. Under pricing inputs, (16) produces a monotone CDF and a nondecreasing convex piecewise-linear put curve; calls follow from put–call parity. A trapezoidal integral of a numerically reconstructed CDF is a different output rule and carries its own integration error, so implementations should state which rule they use.

Before specializing the dependence factorization, we record the numerical error of the finite-lattice portfolio law and, under pricing inputs, its main static consistency properties.

Proposition 1 (Finite-grid error). *Assume the setting of Section 3.1, and let $\varepsilon_h := \frac{dh}{2}$. Then:*

- (i) For every realization $|B_T - B_{T,h}| \leq \varepsilon_h$.
- (ii) For any L_Φ -Lipschitz payoff, $|\mathbb{E}_t[\Phi(B_T)] - \mathbb{E}_t[\Phi(B_{T,h})]| \leq L_\Phi \varepsilon_h$.
- (iii) If F_B and F_{B_h} denote the CDFs of B_T and $B_{T,h}$, then

$$F_B(x - \varepsilon_h) \leq F_{B_h}(x) \leq F_B(x + \varepsilon_h) \quad \forall x \in \mathbb{R}.$$

(iv) If B_T has a density bounded by $\bar{f}_B$, then $\sup_x |F_{B_h}(x) - F_B(x)| \leq \bar{f}_B \varepsilon_h$.

(v) Let Q_B and Q_{B_h} denote the generalized quantile functions of B_T and $B_{T,h}$. Then

$$\|Q_B - Q_{B_h}\|_{L^\infty(0,1)} = \sup_{p \in (0,1)} |Q_B(p) - Q_{B_h}(p)| \leq \varepsilon_h.$$

Proof. See Appendix A. □

Thus every portfolio quantile inherits exactly the same deterministic ε_h accuracy guarantee as the underlying aggregate itself, without any additional regularity assumption on the portfolio density.

Remark 3 (Forward-matched pricing lattice). Under risk-neutral inputs, the continuous portfolio law has the forward mean

$$m_B := \mathbb{E}_t[B_T] = \sum_{r=1}^d w_r \mathbb{E}_t[S_{r,T}],$$

which is fixed by the supplied constituent marginals. Midpoint quantization need not reproduce this mean exactly. If exact forward matching is desired, define

$$\delta_h := m_B - \mathbb{E}_t[B_{T,h}], \quad \tilde{B}_{T,h} := B_{T,h} + \delta_h. \quad (17)$$

Proposition 1(i) implies $|\delta_h| \leq \varepsilon_h$, so

$$\mathbb{E}_t[\tilde{B}_{T,h}] = m_B, \quad |B_T - \tilde{B}_{T,h}| \leq 2\varepsilon_h \quad \text{a.s.}$$

Operationally, the correction shifts every output node by the same δ_h and leaves the contraction and its probability masses unchanged. Strike monotonicity and convexity are therefore preserved exactly, while put–call parity is anchored to the continuous basket forward. The corresponding worst-case Lipschitz and quantile bounds become $2\varepsilon_h$ under this optional correction. If the correction is not applied, the lattice forward mismatch is itself bounded by ε_h and can be reported as part of the numerical discretization error.

Proposition 2 (Pricing consistency and comonotonic bound). *Let C_t be any copula measure and let $F_{1,t}, \dots, F_{d,t}$ be the marginal laws used for pricing, each matching the observed vanilla prices of its constituent at maturity T . Write $\Pi(K) = \mathbb{E}_t[(K - B_T)^+]$ for the undiscounted basket put. Then:*

(i) (Static consistency.) *The construction defines a single joint law at maturity T . Hence put–call parity holds exactly, $K \mapsto \Pi(K)$ is nondecreasing and convex, $\Pi(K) \geq (K - \mathbb{E}_t[B_T])_+$, and every constituent’s vanilla options are repriced exactly. In particular the model admits no static arbitrage among the basket claims and the constituent claims it prices at that maturity.*

(ii) (Convex-order bounds.) *For positive weights and any convex payoff ϕ ,*

$$\phi(\mathbb{E}_t[B_T]) \leq \mathbb{E}_t[\phi(B_T)] \leq \mathbb{E}_t[\phi(B_T^C)], \quad B_T^C := \sum_{r=1}^d w_r Q_{r,t}(V), \quad V \sim \text{U}(0,1), \quad (18)$$

where B_T^C is the comonotonic basket built from the same marginals.

(iii) (Vanilla decomposition of the upper bound.) *For positive weights the comonotonic bound is attained by a static portfolio of constituent options: with $v^* := F_{B^C}(K)$ and $K_r := Q_{r,t}(v^*)$, so that $\sum_r w_r K_r = K$ up to the usual convention at flat parts,*

$$\Pi(K) \leq \Pi^C(K) = \sum_{r=1}^d w_r \mathbb{E}_t[(K_r - S_{r,T})^+]. \quad (19)$$

Proof. See Appendix A. □

Remark 4 (Single-maturity consistency). The no-arbitrage statements in Proposition 2 are cross-sectional statements for the modeled maturity T . Applying the construction separately at several maturities does not by itself impose calendar consistency or generate a martingale dynamics across maturities. Such intertemporal restrictions are outside the single-maturity portfolio-law problem considered here.

For risk-neutral inputs, (18) is a structural guarantee rather than a test of the selected dependence model: every joint law with the supplied marginals lies within the same comonotonic bound. On the lattice, these statements hold up to the error in Proposition 1, while the discrete output rule (16) satisfies the strike-shape constraints exactly. The model-free literature sharpens such bounds when partial dependence information is available (Hobson et al., 2005; Tankov, 2011; Embrechts et al., 2013; Bernard et al., 2017; Puccetti and Wang, 2015; Eckstein et al., 2021; Neufeld et al., 2023). Those methods quantify dependence uncertainty through ranges; our objective is instead to evaluate the full aggregate law under a selected dependence model or scenario.

4 Copula factorization

The factorization (13) is generic; this section specializes it to the dependence class used throughout the paper. We work with vine copulas (Bedford and Cooke, 2001, 2002), which build a d -dimensional copula density from bivariate blocks arranged in a sequence of $d - 1$ linked trees: the first tree couples pairs of uniform margins directly, and each later tree couples conditional margins obtained by applying conditional CDFs (h -functions) along the earlier trees (Aas et al., 2009; Czado, 2019). Two properties make the class a natural fit. Every multivariate density admits a pair-copula decomposition along a vine (Bedford and Cooke, 2002; Czado, 2019), so a vine of arbitrary depth can represent any dependence structure, while truncation depth controls the trade-off between dependence resolution and computational cost, and the granularity of the construction matches that of (13): the density is a product of pair factors, so the d -dimensional dependence-modeling problem decomposes into bivariate ones, each estimable and selectable on its own (Dißmann et al., 2013).

Vines also come with a built-in dimensionality-reduction mechanism. *Truncation* sets every pair copula beyond tree K to independence (Brechmann et al., 2012), cutting the number of pair factors from $d(d - 1)/2$ to $\sum_{j=1}^K (d - j) = Kd - K(K + 1)/2$ —from quadratic to linear in d at fixed K . Because the full-depth vine is exact, the only error comes from the discarded conditional pairs, which are often near independence in practice (Brechmann et al., 2012). In our setting truncation does double duty. Statistically, at fixed depth K it reduces dependence estimation to a linear number of bivariate fits. Computationally, depth one is particularly attractive because every pair factor acts directly on the original ranks. A finite-rank expansion then gives the separable structure of (13) without enlarging the separator state, and the d -dimensional portfolio calculation reduces to the one-dimensional convolution developed below. Higher fixed depths remain compatible with the architecture for plain Bernstein pair copulas, at the cost of larger contraction states; Section 9.2 discusses this extension.

4.1 First-tree copula models

Let $\mathcal{G} = (\{1, \dots, d\}, E)$ be a tree: the vertex set is the set of coordinates, and each edge carries one pair copula. The first-tree copula density is the product of the edge factors (Bedford and Cooke,

2002; Aas et al., 2009),

$$c_{\mathcal{G}}(u) = \prod_{e=(r,s) \in E} c_e(u_r, u_s). \quad (20)$$

The product is again a valid copula density—this is the content of the next proposition.

Proposition 3 (Validity of a first-tree pair-copula product). *If every c_e is a bivariate copula density, then (20) is a d -dimensional copula density.*

Proof. Successive integration of leaf coordinates; see Appendix B. $\square$

The topology of $\mathcal{G}$ is unrestricted. Any spanning tree on the d coordinates therefore defines a valid first-tree dependence model and admits exact leaf-inward sum-product contraction. The familiar D- and C-vines are two extreme cases: a D-vine uses a path, so every interior vertex has degree two, whereas a C-vine uses a star with one vertex of degree $d - 1$. Intermediate trees may combine path-like regions with moderate hubs and can therefore encode economically meaningful dependence structures while retaining exactly $d - 1$ directly modeled pair copulas.

A depth-1 D-vine is the path

$$c_{\text{D}}(u) = \prod_{r=1}^{d-1} c_r(u_r, u_{r+1}), \quad (21)$$

and a depth-1 C-vine with observed root r_0 is the star

$$c_{\text{C}}(u) = \prod_{i \in \mathcal{L}} c_i(u_{r_0}, u_i), \quad \mathcal{L} = \{1, \dots, d\} \setminus \{r_0\}. \quad (22)$$

The leaves of the star are conditionally independent given the observed root.

The path and star also bracket the computational behavior of general first-tree models. The D-vine minimizes the maximum node degree and therefore gives the smallest local contraction states, while the C-vine concentrates all links at one root and produces the largest state among these canonical constructions. General first-tree topologies lie between these cases, with their computational width governed by node degree. We develop the D-vine explicitly in Section 5 because it provides the cleanest and most scalable canonical implementation. The C-vine is treated in Appendix C. The same leaf-inward contraction principle applies to arbitrary first-tree topologies.

Topology is also economically relevant. Since every spanning tree contains exactly $d - 1$ edges, changing the first-tree topology reallocates the directly modeled dependence links rather than increasing their number. The tree can therefore be chosen to retain the dependence relationships most relevant for the portfolio before additional vine depth is introduced.

Depth one is computationally special because all pair copulas act directly on the original ranks, which keeps the contraction state small. At higher vine trees, pair copulas act on h -function transforms involving several original coordinates, so the simple edge-wise contraction developed below no longer applies directly. For plain Bernstein pair copulas, however, these transforms remain polynomial and can be expanded into finite sums of separable factors. The rank-space architecture therefore extends to higher fixed depths, but with larger separator states and correspondingly higher computational cost. We develop the depth-one case as the most tractable implementation and return to higher-depth constructions in Section 9.2.

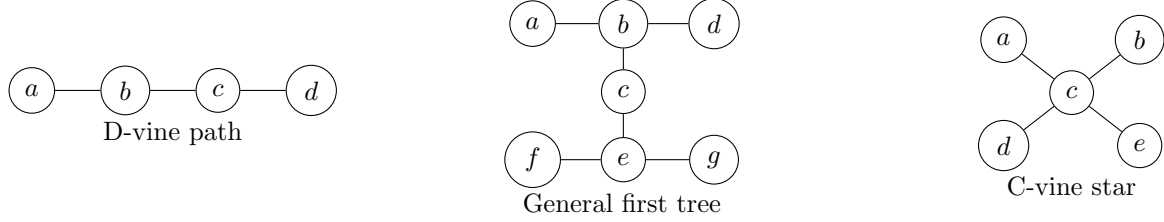


Figure 2: Three first-tree topologies. The D-vine path and C-vine star are the extreme special cases. Arbitrary spanning trees combine elements of them and permit dependence links to be allocated according to the structure of the portfolio.

4.2 Bernstein bivariate copula

For the bivariate building blocks we use Bernstein copulas, constructed from Bernstein polynomials. Besides providing a flexible nonnegative approximation, their algebraic structure is particularly useful for the rank-space architecture: products of Bernstein basis functions remain in the Bernstein family, and the associated h -functions are polynomial. These properties support both the first-tree contractions developed below and the higher-depth extension of Section 9.2. The Bernstein basis of degree n on $[0, 1]$ is the family

$$b_{k,n}(u) = \binom{n}{k} u^k (1-u)^{n-k}, \quad 0 \leq k \leq n, \quad (23)$$

which form a nonnegative partition of unity widely used in approximation because shape properties of the coefficient sequence transfer to the polynomial. Proposition 4 states some important results that will be used throughout the following sections.

Proposition 4 (Bernstein polynomials). *Let $(b_{k,n})_{k=0}^n$ be the n -degree family of Bernstein polynomials. Then the following statements are true:*

- (i) $\int_0^1 b_{k,n}(u) du = \frac{1}{n+1}, \quad \forall k \in \{0, \dots, n\}.$
- (ii) *The integrated basis of each $b_{k,n}$ is $B_{k,n}(u) = \int_0^u b_{k,n}(v) dv = \frac{1}{n+1} \sum_{j=k+1}^{n+1} b_{j,n+1}(u)$, while the integral over an interval $[a, b] \subseteq [0, 1]$ is $\Delta B_{k,n}[a, b] := B_{k,n}(b) - B_{k,n}(a).$*
- (iii) $b_{i,n}(u)b_{j,m}(u) = \omega_{n,m}(i, j)b_{i+j, n+m}(u)$ where $\omega_{n,m}(i, j) = \frac{\binom{n}{i}\binom{m}{j}}{\binom{n+m}{i+j}}.$
- (iv) *For fixed l we have $0 \leq \omega_{n,m}(i, l-i) \leq 1$ and $\sum_i \omega_{n,m}(i, l-i) = 1.$*

Proof. See Appendix B □

Let e be an edge within our vine copula and define n_c as the degree of the Bernstein family, with $R = n_c + 1$. If $A^{(e)} = (A_{ab}^{(e)})_{a,b=0}^{n_c} \in \mathbb{R}_+^{R \times R}$ is a nonnegative coefficient matrix then we define the Bernstein pair copula as

$$c_e(u, v) = \sum_{a=0}^{n_c} \sum_{b=0}^{n_c} A_{ab}^{(e)} b_{a,n_c}(u) b_{b,n_c}(v) = b(u)^\top A^{(e)} b(v), \quad (24)$$

where $b(u) = (b_{0,n_c}(u), \dots, b_{n_c,n_c}(u))^\top$. Uniform margins are equivalent to

$$\sum_b A_{ab}^{(e)} = n_c + 1 \quad \forall a, \quad \sum_a A_{ab}^{(e)} = n_c + 1 \quad \forall b. \quad (25)$$

Thus $A^{(e)}/(n_c + 1)$ is doubly stochastic. The pair CDF and h -function—the conditional CDF $h_{u|v}^{(e)}(u | v) = \partial C_e(u, v)/\partial v$ of the first uniform given the second (Dißmann et al., 2013)—are finite Bernstein sums,

$$C_e(u, v) = \sum_{a,b} A_{ab}^{(e)} B_{a,n_c}(u) B_{b,n_c}(v), \quad (26)$$

$$h_{u|v}^{(e)}(u | v) = \sum_{a,b} A_{ab}^{(e)} B_{a,n_c}(u) b_{b,n_c}(v). \quad (27)$$

Since both B_{a,n_c} and b_{b,n_c} are polynomials, the h -function is itself polynomial; this closure property is what permits higher-tree factors to be re-expanded into finite separable representations.

The contraction itself requires only admissible coefficient matrices $A^{(e)}$ satisfying (25) and is otherwise agnostic about how they are obtained. Section 10 describes the empirical construction of this dependence layer.

5 Exact Contractions

The preceding section allows an arbitrary first-tree topology. We now derive the contraction explicitly for the D-vine path, which has the smallest possible maximum node degree and therefore provides the most scalable canonical implementation. General first trees are contracted by the same leaf-inward sum-product principle, with state sizes determined by their local node degrees. The C-vine star, representing the opposite high-degree extreme, is treated in Appendix C.

5.1 The D-vine as a matrix-product state

The depth-1 D-vine product (21) is a chain of pair factors, each sharing one summation index with its neighbor. Rewriting it as a matrix product makes this index structure explicit: evaluating the dependence density becomes a sequence of small matrix multiplications, and the local rank-cell arrays used to construct the portfolio law inherit the same structure.

Write $b(u) = (b_{0,n_c}(u), \dots, b_{n_c,n_c}(u))^T$ and denote the coefficient matrix on edge $(r, r + 1)$ by $A^{(r)}$. Define

$$L_1(u)_j := [b(u)^T A^{(1)}]_j, \quad (28)$$

$$G_r(u)_{ij} := b_{i,n_c}(u) [b(u)^T A^{(r)}]_j, \quad 2 \leq r \leq d - 1, \quad (29)$$

$$R_d(u)_i := b_{i,n_c}(u). \quad (30)$$

Theorem 3 (Continuous D-vine matrix-product identity). *The first-tree D-vine density satisfies*

$$c_D(u) = L_1(u_1)^T G_2(u_2) \cdots G_{d-1}(u_{d-1}) R_d(u_d). \quad (31)$$

The bond dimension (the tensor-train rank) is $R = n_c + 1$ and does not depend on the marginal representation.

Proof. The expansion over bond indices regroups the pair factors; see Appendix D. $\square$

In the language of tensor networks, $L_1(u_1)^T$ is a $1 \times R$ row vector, each $G_r(u_r)$ an $R \times R$ matrix (a *core*), and $R_d(u_d)$ an $R \times 1$ column vector: (31) is a matrix-product state of bond dimension R . The bond index j_r is the shared Bernstein index of edge $(r, r + 1)$: dependence propagates along the chain through exactly R states, which is why the contraction width is independent of d . This differs

from tensor-train approaches that numerically compress a model-specific characteristic-function or pricing tensor (Kastoryano and Pancotti, 2022; Arenstein and Kastoryano, 2026). Here the tensor-train representation is exact for the chosen first-tree Bernstein dependence model, and its bond dimension $R = n_c + 1$ is known a priori. Moreover, the cores represent dependence in rank space, so changing a marginal quantile map leaves the dependence train unchanged.

5.2 D-vine: analytic local cell matrices in the plain basis

Let $[u_{r,k}^-, u_{r,k}^+]$ be the rank interval associated with local portfolio-contribution cell k . Define the exact cell arrays

$$\mathbf{L}_{1,j,k} := \int_{u_{1,k}^-}^{u_{1,k}^+} L_1(u)_j \, du, \quad (32)$$

$$\mathbf{G}_{r,ij,k} := \int_{u_{r,k}^-}^{u_{r,k}^+} G_r(u)_{ij} \, du, \quad (33)$$

$$\mathbf{R}_{d,i,k} := \int_{u_{d,k}^-}^{u_{d,k}^+} R_d(u)_i \, du. \quad (34)$$

Using Proposition 4,

$$\mathbf{L}_{1,j,k} = \sum_{a=0}^{n_c} A_{aj}^{(1)} \Delta B_{a,n_c}[u_{1,k}^-, u_{1,k}^+], \quad (35)$$

$$\mathbf{G}_{r,ij,k} = \sum_{a=0}^{n_c} A_{aj}^{(r)} \omega_{n_c,n_c}(i, a) \Delta B_{i+a,2n_c}[u_{r,k}^-, u_{r,k}^+], \quad (36)$$

$$\mathbf{R}_{d,i,k} = \Delta B_{i,n_c}[u_{d,k}^-, u_{d,k}^+]. \quad (37)$$

These formulas contain no marginal-density coefficients and no numerical integration. The marginal input appears only through the rank endpoints.

5.3 D-vine: discrete convolution and Fourier contraction

Theorem 2 expresses the mass array of the quantized portfolio law as discrete convolutions of one-dimensional cell arrays. Convolution is diagonalized by the Fourier transform, so these convolutions become pointwise products of transformed arrays—precisely the setting in which the fast Fourier transform applies. Pad every cell array to a common length

$$L \geq M_\Sigma; \quad (38)$$

this is the no-wrap condition for linear (as opposed to circular) convolution, and choosing an FFT-friendly L changes runtime but not the discrete model. Let a hat denote the length- L discrete Fourier transform along the cell index. The transform of the mass array is evaluated frequency by frequency as

$$\widehat{\pi}(\xi) = \widehat{\mathbf{L}}_1(\xi)^\top \widehat{\mathbf{G}}_2(\xi) \cdots \widehat{\mathbf{G}}_{d-1}(\xi) \widehat{\mathbf{R}}_d(\xi), \quad (39)$$

where each factor is the transform of a cell array (32)–(34). One inverse transform of the product yields the complete mass array $\{\pi_n^{(h)}\}$ in (15). This array determines the full quantized portfolio law and hence all one-dimensional functionals derived from it. Under pricing inputs, the entire strike curve follows without an additional contraction. The transform uses complex signed arithmetic, but the underlying discrete masses are nonnegative.

Algorithm 1: Full plain-basis D-vine portfolio-law build

1. Choose h , compute a_r , W_r , M_r , and a padding length $L \geq M_\Sigma$.
2. For every local cell, map its contribution interval to the rank interval $[u_{r,k}^-, u_{r,k}^+]$ using the marginal CDF or quantile table.
3. Assemble $\mathbf{L}_1$, $\mathbf{G}_2, \dots, \mathbf{G}_{d-1}$, and $\mathbf{R}_d$ from (35)–(37).
4. Zero-pad and transform every scalar cell array along the aggregate-cell axis.
5. Evaluate (39) at every frequency and apply the inverse transform.
6. Record signed FFT residuals, project only roundoff-scale negative masses if necessary, normalize only after reporting the mass correction, and construct the portfolio mass array and derived outputs on the nodes (12).

6 Polynomial Complexity and Comparison with Alternatives

This section collects the cost of the constructions of Section 5, states the complexity results that motivate the paper, and compares the architecture with two natural computational baselines. For comparison, a direct tensor grid with M states per coordinate requires $O(M^d)$ joint states before aggregation and therefore scales exponentially in the portfolio dimension. The contractions below avoid this d -dimensional state grid and operate instead on the one-dimensional aggregate lattice.

6.1 Assembly, contraction, and memory costs

For the D-vine, exact cell assembly by (36) costs conservatively $O(\sum_r M_r R^3)$ if each cell uses a dense matrix multiplication. Midpoint sampling reduces that stage to $O(\sum_r M_r R^2)$ but replaces the exact cell masses by a quadrature with error $O(h^2)$ per cell (a midpoint rule on a smooth one-dimensional integrand); that local error is then separate from, and additional to, the portfolio quantization error of Proposition 1. The FFT and frequency contractions cost

$$O(dR^2 L \log L + dR^2 L), \quad (40)$$

and a streaming implementation requires $O(R^2 L)$ working memory. A direct cell-space implementation replaces each Fourier product by an ordinary convolution; its conservative cost is

$$O\left(R^2 \sum_{r=2}^d P_{r-1} M_r\right), \quad P_{r-1} := 1 + \sum_{j < r} (M_j - 1), \quad (41)$$

which is $O(R^2 L^2)$ in the worst case. The direct path is positivity-preserving and can be faster on very short arrays, whereas the FFT path dominates beyond an empirically determined crossover. Because the FFT path uses signed complex arithmetic, implementations should report the signed inverse-transform residual and any negative roundoff mass before projection; the direct path serves as the reference.

For the C-vine, the plain leaf assembly costs $O(\sum_i M_i R^2)$ and the root assembly costs $O(M_0 R_\star)$. The transform and contraction costs are

$$O((dR + R_\star)L \log L + dRR_\star L), \quad (42)$$

with streaming memory $O(R_\star L)$ in addition to the current leaf slab. The direct coefficient-axis recurrence has the same asymptotic operation count as the earlier binomially rescaled construction, but materially smaller intermediate dynamic range.

6.2 Polynomial cost in the basket dimension

Fix a common per-constituent resolution, so that $M_r \leq M$ for all r and the common aggregate lattice has length $L = O(dM)$ by (12). Write $R = n_c + 1$ for the pair bond dimension and $R_\star = (d-1)n_c + 1$ for the C-vine root state.

Corollary 1 (Polynomial cost in the basket dimension). *Under the plain-basis contractions of Section 5 and with $L = O(dM)$, the D-vine build and contraction (40) and the C-vine build and contraction (42) cost*

$$(D\text{-vine}) \quad O(dR^2 L \log L) = O(d^2 n_c^2 M \log(dM)), \quad (43)$$

$$(C\text{-vine}) \quad O((dR + R_\star)L \log L + dR R_\star L) = O(d^3 n_c^2 M + d^2 n_c M \log(dM)), \quad (44)$$

respectively. Both are polynomial in the basket dimension d for fixed pair degree n_c and resolution M .

Equations (43)–(44) hold the per-constituent resolution M —equivalently the spacing h —fixed as d grows. Because the worst-case quantization bound of Proposition 1 is $\varepsilon_h = dh/2$, that convention lets the guaranteed accuracy degrade linearly in d . The following restatement fixes the accuracy instead and is the one to quote.

Corollary 2 (Cost at fixed accuracy). *Fix a target worst-case quantization error $\varepsilon > 0$ and let the contribution widths W_r be bounded uniformly in d . Then $h = 2\varepsilon/d$, $M = O(d/\varepsilon)$ and $L = O(d^2/\varepsilon)$, so the builds cost*

$$(D\text{-vine}) \quad O\left(\frac{d^3 n_c^2}{\varepsilon} \log \frac{d^2}{\varepsilon}\right), \quad (C\text{-vine}) \quad O\left(\frac{d^4 n_c^2}{\varepsilon}\right). \quad (45)$$

Both remain polynomial in d .

Remark 5 (Worst case versus realized quantization error). The bound $\varepsilon_h = dh/2$ adds the d per-coordinate residuals at their extremes. Each residual $Z_r - D_{r,h}(Z_r)$ lies in $[-h/2, h/2]$ and, as a function of u_r , oscillates on the scale of a single cell; treated as approximately uncorrelated and uniform, the aggregate has standard deviation $h\sqrt{d/12} \approx 0.29 h\sqrt{d}$ rather than $0.5hd$ —a factor of eight already at $d = 20$. Holding the realized root-mean-square error fixed therefore requires only $h \propto \varepsilon/\sqrt{d}$, and the D-vine cost in (45) improves to $O(d^{2.5} n_c^2 \varepsilon^{-1} \log(\cdot))$. The worst-case bound remains the guarantee.

6.3 Sampling error and tail quantiles

Monte Carlo is the natural stochastic benchmark. Generating and evaluating N d -dimensional portfolio paths costs $O(Nd)$, while its absolute error decays at the dimension-free rate $N^{-1/2}$. The difficulty arises instead from relative precision in rare tails. Fix a relative moneyiness $\delta \in (0, 1)$ and consider

$$p_d = \Pr(B_T \leq (1 - \delta)\mathbb{E}_t[B_T])$$

for an equally weighted portfolio. Under standard large-deviation conditions,

$$p_d = \exp\{-dI(\delta) + o(d)\}, \quad I(\delta) > 0. \quad (46)$$

A crude Monte Carlo estimator of p_d has absolute and relative standard errors

$$\sqrt{\frac{p_d(1-p_d)}{N}} \quad \text{and} \quad \sqrt{\frac{1-p_d}{Np_d}},$$

respectively. Thus absolute probability error decreases as the event becomes rarer, while maintaining fixed *relative* precision requires

$$N \gtrsim p_d^{-1} = \exp\{dI(\delta) + o(d)\}. \quad (47)$$

A similar relative-precision problem can arise for option values and other tail functionals whose expectation is concentrated on rare states. Importance sampling can improve this trade-off, but requires a tilt tailored to the relevant joint law and tail quantity.

The contraction is sampling-free and provides a uniform horizontal error guarantee for the portfolio law. Proposition 1 gives

$$F_B(x - \varepsilon_h) \leq F_{B_h}(x) \leq F_B(x + \varepsilon_h),$$

and therefore

$$\sup_{p \in (0,1)} |Q_B(p) - Q_{B_h}(p)| \leq \varepsilon_h.$$

Every portfolio quantile, including deep-tail quantiles, is therefore located within the same deterministic distance ε_h , without any regularity assumption on the portfolio density. If a vertical CDF bound is desired, the additional assumption that the portfolio density is bounded by $\bar{f}_B$ yields

$$\sup_x |F_{B_h}(x) - F_B(x)| \leq \bar{f}_B \varepsilon_h.$$

This horizontal guarantee is particularly useful for tail evaluation. Under the usual fixed- p regularity conditions, the asymptotic standard error of a Monte Carlo sample quantile is proportional to

$$\frac{\sqrt{p(1-p)}}{\sqrt{N} f_B(Q_B(p))},$$

and hence depends on both the number of simulated paths and the local portfolio density. Rank-space contraction instead controls the location of the entire quantile function directly and uniformly in p .

The same contraction produces the complete one-dimensional portfolio mass array. The full CDF, quantiles, risk measures, and, under pricing inputs, the entire strike curve are therefore obtained from a single calculation with the same numerical error budget. Moreover, the offline/online decomposition is retained under changing marginals: a marginal update requires rebuilding only the corresponding local factor against the cached contraction environment, rather than reconstructing the high-dimensional joint calculation.

7 Tail Completion, Clipping, and the Warped Basis

The core results of Sections 2–5 assume bounded supports and the plain Bernstein basis. This section treats the two practical issues that follow: how unbounded tails are completed or truncated, and how the finite-level tail resolution of the basis can be improved by warping. At any fixed Bernstein degree, the pair-copula densities remain bounded and hence asymptotically tail-independent. Our tail analysis therefore concerns dependence at economically relevant finite probability levels and the role of warping in reallocating resolution toward the corners.

7.1 Tail completion, truncation, and clipping

The rank formulation supports three consistent tail treatments: a continuous parametric completion; an unbounded marginal with a wide numerical truncation and a separately controlled residual; and coordinatewise clipping, which converts each omitted tail into a boundary atom represented by a rank interval. For clipping, define $S_r^{\text{clip}} = \min\{\bar{s}_r, \max\{S_r, \underline{s}_r\}\}$ and $B^{\text{clip}} = \sum_r w_r S_r^{\text{clip}}$.

Proposition 5 (Tail first-moment bound). *For any L_Φ -Lipschitz payoff,*

$$\begin{aligned} & \left| \mathbb{E}_t[\Phi(B_T)] - \mathbb{E}_t[\Phi(B_T^{\text{clip}})] \right| \\ & \leq L_\Phi \sum_{r=1}^d |w_r| \mathbb{E}_t \left[(\underline{s}_r - S_{r,T})_+ + (S_{r,T} - \bar{s}_r)_+ \right]. \end{aligned} \quad (48)$$

Proof. Lipschitz continuity and the triangle inequality; see Appendix A. $\square$

For unbounded Lipschitz functionals, including calls and puts, (48) supplies the appropriate diagnostic: a tail probability alone does not control the contribution of omitted states to the functional. Under risk-neutral inputs, any completed marginal must additionally match the forward under the selected numeraire. Clipping generally changes that mean and is therefore a numerical sensitivity device unless its boundary atoms or tail parameters are adjusted consistently.

7.2 Corner ceilings and the role of warping

A plain Bernstein pair copula is a finite, nonnegative, doubly-stochastic mixture of basis products. This buys positivity and analytic rank-cell masses, but it also caps the sharpness of the corners: the density is bounded and both lower and upper joint tail probabilities are $O(q^2)$ as $q \downarrow 0$ (Proposition 6), so a plain Bernstein copula is asymptotically tail-independent at every finite degree. For equity portfolios under stress, where joint downside co-movement is the object of interest, this is a limitation of the basis rather than of the architecture; raising the degree sharpens the interior fit but does not create asymptotic tail dependence.

The warped basis is a *resolution* remedy, not a tail-dependence remedy. It composes the Bernstein basis with a smooth density warp on each margin, concentrating resolution where tail mass must be captured at finite levels q , while preserving the finite-rank pair factorization the contraction needs. It does not create asymptotic tail dependence: the corner ceilings persist for every fixed smooth warp (Proposition 6), and genuine tail dependence would require leaving the finite-rank Bernstein class altogether. The cost of warping is twofold and is made explicit below: analytic rank-cell masses are generally lost (one-dimensional integration returns), and a common warp at a C-vine root introduces a dynamic-range factor that must be controlled.

7.3 Warped Bernstein pair copulas

Let

$$\nu(x) = \sum_{a=0}^{n_c} \nu_a b_{a,n_c}(x), \quad \nu_a \geq 0, \quad \sum_{a=0}^{n_c} \nu_a = n_c + 1, \quad (49)$$

with $\nu(x) \geq \nu_{\min} > 0$ on $[0, 1]$. Then ν is a density. Let $V(x) = \int_0^x \nu(z) dz$ and $\Lambda_\nu = V^{-1}$. Define

$$\varphi_a(u) := b_{a,n_c}(\Lambda_\nu(u)) \Lambda'_\nu(u), \quad \Lambda'_\nu(u) = \frac{1}{\nu(\Lambda_\nu(u))}. \quad (50)$$

Each φ_a integrates to $1/(n_c + 1)$ and $1 = \sum_a \nu_a \varphi_a(u)$. Therefore

$$c_e(u, v) = \sum_{a,b} A_{ab}^{(e)} \varphi_a(u) \varphi_b(v) \quad (51)$$

has uniform margins if and only if

$$\sum_b A_{ab}^{(e)} = (n_c + 1) \nu_a, \quad \sum_a A_{ab}^{(e)} = (n_c + 1) \nu_b. \quad (52)$$

The independence copula is $A^{(e)} = \nu \nu^\top$.

Proposition 6 (Finite-level corner ceilings). *For a warped pair copula satisfying (49)–(52),*

$$c_e(0, 0) \leq \frac{n_c + 1}{\nu_0}, \quad c_e(1, 1) \leq \frac{n_c + 1}{\nu_{n_c}}, \quad (53)$$

$$c_e(0, 1), c_e(1, 0) \leq \frac{n_c + 1}{\max\{\nu_0, \nu_{n_c}\}}. \quad (54)$$

Consequently, the lower-left and upper-right joint tail probabilities are $O(q^2)$ as $q \downarrow 0$ for every fixed smooth warp.

Proof. Endpoint evaluation of the warped basis; see Appendix E. $\square$

Warping preserves a finite-rank pair factorization but generally removes analytic rank-cell masses: the cell tensors of Section 5.1 and Appendix C.2 are then evaluated by one-dimensional quadrature or a pretabulated antiderivative.

7.4 The warped C-vine root: cell arrays and dynamic range

For a warped C-vine, all incident edges must share a common root-side warp ν . Each incident root basis function is then $b_{a,n_c}(\Lambda_\nu(u_0)) \Lambda'_\nu(u_0)$; the coefficient recursion (75) is unchanged and the exact continuous identity becomes

$$c_C(u) = \{\Lambda'_\nu(u_0)\}^m \sum_{A=0}^{mn_c} b_{A,mn_c}(\Lambda_\nu(u_0)) H_A^{(m)}(u_1, \dots, u_m), \quad (55)$$

where each leaf $g_{i,a}$ uses its leaf-side warped basis. The root cell arrays (79) are replaced by

$$\mathbf{R}_{A,k}^{\text{warp}} := \int_{u_{0,k}^-}^{u_{0,k}^+} \{\Lambda'_\nu(u)\}^m b_{A,N_\star}(\Lambda_\nu(u)) du, \quad (56)$$

which remains one-dimensional but is not generally a Bernstein antiderivative.

Even normalized coefficient recursion cannot remove the intrinsic factor $\{\Lambda'_\nu(u)\}^m$ in (56). Since $1/\nu_{\max} \leq \Lambda'_\nu(u) \leq 1/\nu_{\min}$, its range is bounded by

$$\left(\frac{\nu_{\max}}{\nu_{\min}} \right)^m. \quad (57)$$

A warped C-vine should therefore enforce a non-negligible $\nu_{\min}$ and rescale root states before Fourier closure; the scaling and the associated stability checks are given in Appendix H.

8 Online Marginal Updates and Cache Semantics

A marginal update changes the map from portfolio-contribution cells to rank intervals and therefore changes one local D-vine core or one C-vine leaf/root slab. The word *local* has two distinct meanings.

Definition 1 (Baseline-relative update). A baseline-relative bump evaluates one changed marginal against contraction environments cached from a fixed baseline and then discards the bump.

Definition 2 (Persistent update). A persistent update accepts the changed marginal as the new baseline. All cached environments that cross the changed factor must then be refreshed or maintained by a dynamic product structure.

Remark 6 (What is, and is not, dimension-free). The *number of factors* recomputed by a baseline-relative bump is one, whatever d is; this is the sense in which the online layer does not degrade as constituents are added. It is not a claim that the cost is a constant: the common lattice length obeys $L = O(dM)$ by (12), and the C-vine root state $R_\star$ grows linearly in d , so the bump costs of (59) and (61) inherit that growth. Nor does it extend to persistent updates, which invalidate every environment crossing the changed coordinate and cost $O(dR^2L)$ in the straightforward scheme. Reported timings must state which of the two regimes they measure.

8.1 D-vine update costs

At each Fourier frequency, let $\vec{V}_{r-1}$ be the row-vector product through coordinate $r-1$ and let $\overleftarrow{V}_{r+1}$ be the column-vector product from coordinate $r+1$ to the end. Replacing one interior core $\hat{G}_r$ gives

$$\hat{\pi}_{\text{bump}}(\xi) = \vec{V}_{r-1}(\xi) \hat{G}_r^{\text{new}}(\xi) \overleftarrow{V}_{r+1}(\xi). \quad (58)$$

After local cell assembly, a baseline-relative update therefore costs

$$O(R^2L \log L + R^2L). \quad (59)$$

If the update is accepted and every prefix and suffix vector is stored, a straightforward refresh costs $O(dR^2L)$ after the new transform, because all prefixes to the right and suffixes to the left are stale. A segment tree of frequency-wise matrix products instead supports an accepted core update in

$$O(R^2L \log L + R^3L \log d) \quad (60)$$

with $O(dR^2L)$ tree memory. Which scheme is preferable depends on R , d , and the frequency of accepted updates.

8.2 C-vine update costs

If a leaf-exclusion aggregate for leaf i is cached relative to a fixed baseline, a new leaf slab can be combined with it at cost

$$O(RL \log L + RR_\star L). \quad (61)$$

A root update leaves the leaf aggregate unchanged and costs

$$O(R_\star L \log L + R_\star L). \quad (62)$$

After accepting a leaf update, every other leaf-exclusion environment is stale. The straightforward persistent strategy rebuilds the complete aggregate at cost $O(dRR_\star L)$ after the leaf transform. A balanced product tree can reduce the number of updated nodes, but a direct coefficient-axis convolution near the root still has a cost of order $O(R_\star^2L)$; coefficient-axis FFTs or structured compression are separate extensions rather than part of the headline bound. Table 1 summarizes the two regimes, and Algorithm 2 gives the bump procedure.

Operation	Baseline-relative work	Work after accepting the update
D-vine interior marginal	$O(R^2 L \log L + R^2 L)$	$O(dR^2 L)$ prefix/suffix refresh, or $O(R^3 L \log d)$ segment-tree refresh
C-vine leaf marginal	$O(RL \log L + RR_* L)$ with an exclusion state	$O(dRR_* L)$ straightforward aggregate rebuild
C-vine root marginal	$O(R_* L \log L + R_* L)$	same; leaf aggregate remains valid
Location-only shift	translate κ	no factor refresh
Width or padding change	not local	full lattice and transform rebuild
Cached output query	$O(1)$ interpolation for grid-indexed outputs such as strike values	otherwise inverse transform and cumulative reconstruction

Table 1: One-shot bump costs versus persistent cache maintenance. The length L and the C-vine state R_* may themselves grow with the basket dimension.

8.3 Geometry and output caches

An update may also change the aggregate geometry. A common location shift changes only κ and translates the strike axis. A change in a support width, portfolio weight, or tail completion changes W_r and possibly M_r and L ; if so, all padded transforms are invalid. Some implementations quantize widths to integer multiples of h to stabilize cache keys. That is an additional approximation and must be included in the error budget.

Evaluation of retained one-dimensional outputs is cheap only after the final mass, CDF, and any required integrated-payoff arrays have been cached. In particular, strike interpolation is $O(1)$ on a uniform grid once the relevant pricing arrays are retained. If the implementation reconstructs the inverse FFT and cumulative arrays on every query, the query cost includes that reconstruction and should not be reported as $O(1)$.

Algorithm 2: Baseline-relative one-coordinate bump

1. Preserve the baseline local factors and the two environments that meet at the changed coordinate.
2. Recompute only that coordinate’s rank-cell endpoints and local cell tensor, provided h , M_r , and L remain unchanged.
3. Transform the new local tensor and contract it with the two baseline environments.
4. Invert the transformed output and reconstruct the bumped portfolio law and required functionals. If the bump is discarded, the baseline caches remain valid.
5. If the bump is accepted, refresh every stale environment or update a dynamic product tree before processing the next accepted change.

9 Extensions and Generalizations

The depth-one D- and C-vines developed above are computationally particularly convenient, but neither the path and star topologies nor depth one are fundamental restrictions of the rank-space architecture. This section describes how the construction of the portfolio law extends beyond these cases and where the additional computational cost enters.

9.1 General first-tree topologies

Proposition 3 applies to an arbitrary spanning tree, and sum-product contraction is exact on any such tree. The D-vine path and C-vine star are therefore two special cases of a general leaf-inward message-passing construction.

If coordinate r has graph degree δ_r , its rank enters δ_r pair factors. Repeated application of Proposition 4(iii) collapses their product to a Bernstein polynomial of degree $\delta_r n_c$, so the local polynomial state has size $\delta_r n_c + 1$. An interior D-vine node has $\delta_r = 2$, whereas the C-vine root has $\delta_r = d - 1$; general trees interpolate between these cases. The number of modeled pair links remains $d - 1$, while contraction width is governed by the distribution of node degrees.

This flexibility permits the first tree itself to be used as an accuracy-cost design variable. Path-like regions can be combined with moderate hubs, for example to represent sectoral dependence while connecting sectors through a small number of broader market links. At fixed first-tree complexity, topology therefore reallocates the available dependence links toward the relationships that matter most for the portfolio. Only when this is insufficient need additional vine depth be introduced.

9.2 Higher vine depth

Depth-one truncation is the cheapest implementation of the architecture rather than a fundamental boundary. At higher vine trees, pair copulas act on conditional ranks obtained from h -functions and therefore depend on several original rank coordinates. The simple edge-wise states used at depth one no longer suffice.

For plain Bernstein pair copulas, however, the h -function in (27) is itself a polynomial. Composing another Bernstein pair copula with such conditional ranks therefore produces a finite multivariate polynomial, which can again be expanded as a finite sum of products of univariate functions. The same rank-space and convolution logic can consequently be used to construct the portfolio law after enlarging the contraction state.

What grows rapidly with depth is the width of this representation. At depth two, for example, factors involve triples of original coordinates and eliminating one coordinate generally leaves messages on pairs rather than on single coordinates. More generally, a vine truncated at fixed depth K leads to higher-dimensional separators. For structured topologies such as truncated D- and C-vines, the separator size remains bounded when K is fixed, so contraction remains polynomial in the portfolio dimension d . Its dependence on K and on the Bernstein degree, however, becomes increasingly expensive. Depth therefore provides a trade-off between dependence resolution and computational cost rather than a binary choice between tractability and generality.

Taken together, topology, polynomial degree, and vine depth provide complementary ways to allocate computational effort. At fixed depth, the topology and ordering can be chosen to retain the dependence relations most relevant for the portfolio without increasing the number of modeled pair links; within a fixed topology class, such as a D-vine, this can improve approximation quality without changing the asymptotic contraction cost. Increasing the Bernstein degree then refines the representation of the retained pairwise dependence. Only when these adjustments are insufficient is additional vine depth required, at the cost of larger separator states. The architecture therefore exposes a direct trade-off between dependence resolution and computational scalability that can be chosen according to the dimension and accuracy requirements of the application.

The exact polynomial closure is specific to the plain Bernstein construction. The warped basis of Section 7.3 retains the finite-rank factorization at depth one but does not preserve polynomial closure under repeated h -function composition. Higher-depth warped vines would therefore require an additional separable or low-rank approximation of the composed factors. Developing such compression

schemes and corresponding error bounds is a natural extension of the present architecture.

10 Construction of the Dependence Layer

The preceding analysis treats the vine topology and Bernstein coefficient matrices as fixed offline inputs. The contraction itself is agnostic about how these inputs are obtained. They may be estimated directly in rank space or constructed as a tractable representation of a richer dependence model.

For empirical applications, we use a teacher–student construction. A flexible teacher copula is first fitted to data or otherwise specified under the relevant measure. A first-tree topology is then selected, and the teacher is distilled offline into a Bernstein vine by matching pairwise rank-cell probabilities subject to the copula-margin constraints. The Bernstein degree is increased until the resulting student reproduces the teacher sufficiently closely in finite-level tail dependence and in the portfolio laws relevant for the application. Appendix I gives the construction and validation procedure in detail.

This separation is also important for interpreting the error guarantees. Teacher specification and Bernstein distillation determine the dependence-model and representation error. Conditional on the resulting marginal and dependence inputs, the lattice, tail-treatment, quadrature, and floating-point errors are numerical and are controlled separately by the results above. Appendix I summarizes the corresponding error budget.

11 High-Dimensional Accuracy and Runtime Benchmark

We benchmark the complete rank-space pipeline in a controlled high-dimensional setting in which the data-generating dependence structure is known. For $d \in \{10, 50, 100\}$, we generate a Gaussian copula whose conditional-independence graph has a hierarchical first-tree topology. The topology is designed to resemble a broad-to-local dependence organization, such as market-wide, macro, sector, industry, and asset groupings, although these labels are purely mnemonic in the simulation. Formally, every node is simply one copula coordinate, and the correlation between two nodes is the product of the edge loadings along their unique connecting path. The same $d - 1$ edges are supplied as the first tree of the fitted Bernstein vine. By construction, the benchmark is therefore structurally in-model: it tests the finite Bernstein representation, the numerical implementation, and the resulting portfolio-law calculation without introducing first-tree misspecification.

Figure 3 illustrates the hierarchical first-tree topology for the $d = 100$ realization. It consists of one central root, three primary hubs, ten secondary hubs, nineteen tertiary hubs, and sixty-seven terminal nodes. All 100 nodes are treated as observed portfolio constituents in the benchmark, and all pair copulas shown in the figure belong to the same first vine tree. The example therefore illustrates that a first-tree representation can already encode a substantially nested dependence backbone, rather than being restricted to a path or a single hub-and-spoke structure.

For every dimension, the pair copulas are fitted on a fixed 1024×1024 rank grid using 10^8 simulated copula draws and a common Bernstein degree $n_c = 30$. The basis is unwarped. The fitted dependence representation is then passed to the rank-space FFT implementation for an equally weighted basket with $w_i = 1/d$ and synthetic marginals. The marginals have common mean 100, heterogeneous scales, and finite supports constructed from renormalized normal CDF tables. The aggregate lattice spacing is fixed at 2^{-10} across dimensions. Accuracy is assessed against an independent stream of 10^8 Monte Carlo draws from the original copula, mapped through the same

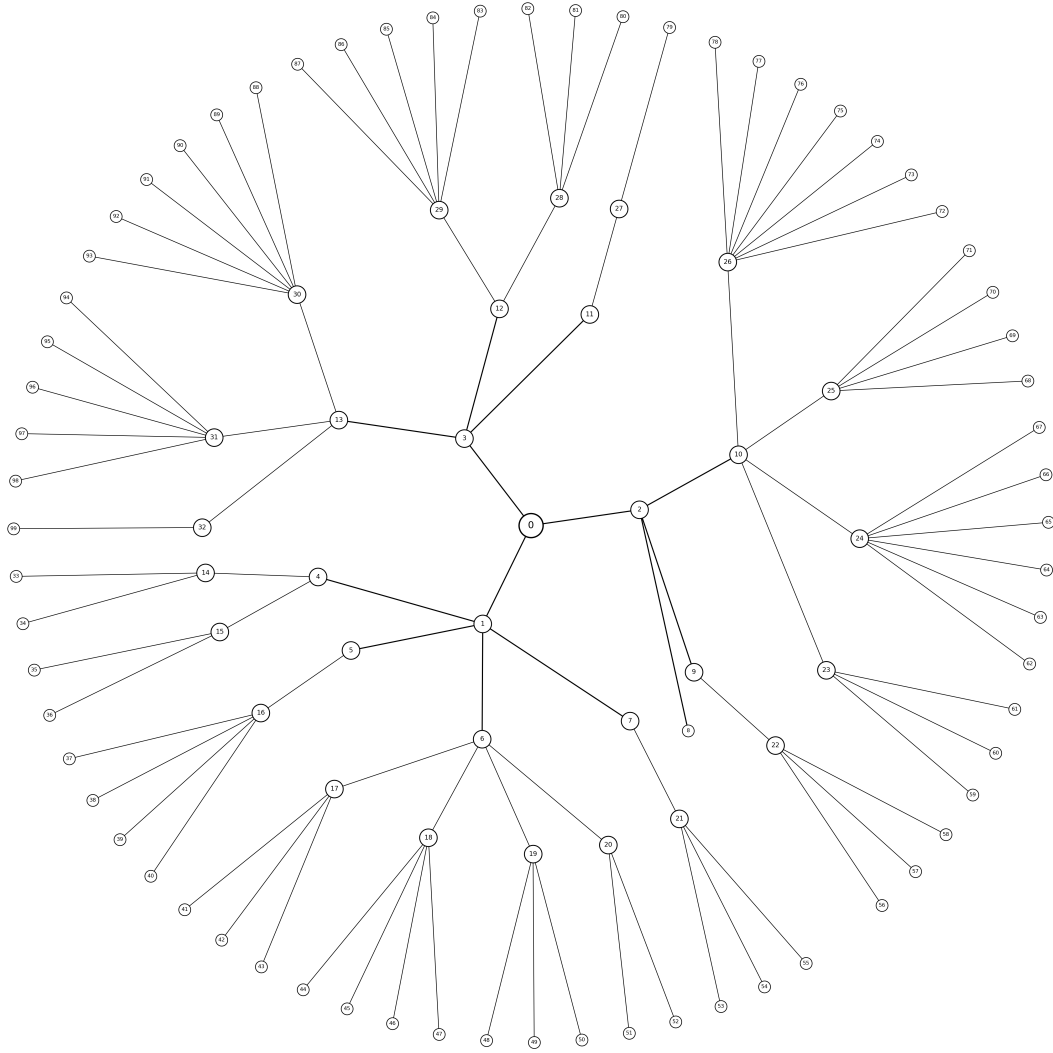


Figure 3: First-tree topology used in the $d = 100$ benchmark. Each node is an observed portfolio coordinate and each edge is one fitted first-tree pair copula.

d	Offline fit (s)	Full-law evaluation (ms)	10^8 -path MC (s)	Δ_F
10	8.75	5.35	2.68	5.06×10^{-4}
50	11.68	33.64	5.95	3.99×10^{-4}
100	13.83	92.99	9.98	9.82×10^{-4}

Table 2: High-dimensional benchmark results. “Offline fit” includes streaming pair-histogram construction and Bernstein fitting. “Full-law evaluation” is the mean runtime of repeated compiled evaluations of the complete portfolio CDF. The Monte Carlo column reports the runtime of the independent 10^8 -path reference calculation. The final column is the maximum absolute CDF discrepancy on the common support grid.

marginal tables. We report

$$\Delta_F := \max_k \left| F_{\text{RS}}(x_k) - \hat{F}_{\text{MC}}(x_k) \right|,$$

where the maximum is taken over all unpadded deterministic support points.

All calculations were performed on NVIDIA RTX 3090 GPUs with 24 GB of memory, with each dimension processed entirely on a single GPU. The dependence fit streams the simulated observations and retains only the $d - 1$ pairwise rank histograms. At $d = 100$ these occupy approximately 396 MB. Reported evaluation times refer to repeated calls after JAX compilation.

Table 2 shows that repeated evaluation of the complete portfolio law remains fast as dimension increases. At $d = 100$, the full portfolio CDF is evaluated in about 93 milliseconds. Across all three reported dimensions, the maximum absolute discrepancy from the independent 10^8 -path Monte Carlo reference remains below 10^{-3} . The corresponding reference Monte Carlo calculation takes about 10 seconds at $d = 100$, while dependence estimation remains an offline operation. The Monte Carlo calculation serves here as a high-precision external reference rather than as the basis for a same-error runtime comparison. Likewise, Δ_F is an empirical benchmark statistic and does not replace the deterministic numerical bounds established above.

The benchmark deliberately isolates a setting in which the first-tree conditional-independence structure is correctly specified. Real portfolio dependence may contain additional conditional relationships that require deeper vine trees. As discussed in Section 9.2, these can be incorporated by increasing vine depth, with larger separator states and higher computational cost. The benchmark nevertheless illustrates that restricting the representation to the first vine tree need not imply a simple dependence topology: that tree can itself encode a substantially nested hierarchy of broad and local co-movements. For aggregate risk, and in particular at finite tail levels where joint movements are often dominated by broad common dependence, such a backbone may already capture much of the economically relevant structure even when additional conditional dependencies remain. The adequacy of depth one is ultimately an empirical question, and additional vine depth provides the corresponding accuracy–runtime trade-off when those residual dependencies matter. In more general models, internal nodes could also represent explicitly modeled non-traded coordinates rather than portfolio constituents; the present benchmark takes the cleaner case in which every node is observed and investable.

12 Numerical Applications

We now illustrate two financial applications of the portfolio law. The first uses comonotonicity to obtain a sharp dependence-robust pricing bound with a direct interpretation for margin requirements.

The second shows how full-law comparisons can distinguish portfolios that are equivalent under a variance-based allocation criterion.

12.1 Comonotonic dependence bounds and margin benchmarks

Conservative upper bounds matter whenever decisions must remain robust to dependence misspecification. This includes conservative bounds on portfolio tail risk and, perhaps most directly in practice, the determination of margin requirements for portfolio derivatives. For long-only portfolios, the separation of marginals and dependence in our architecture makes such bounds directly available.

Holding the constituent marginals fixed, consider the comonotonic coupling

$$B_T^C = \sum_{i=1}^d w_i Q_{i,t}(V), \quad V \sim U(0,1), \quad w_i \geq 0. \quad (63)$$

All constituents share the same rank and therefore

$$Q_{B^C}(q) = \sum_{i=1}^d w_i Q_{i,t}(q). \quad (64)$$

The comonotonic portfolio is maximal in convex order among all joint laws with the same marginals (Dhaene et al., 2002b,a). It therefore provides a conservative dependence bound for convex portfolio functionals, including corresponding tail-risk measures under physical or scenario inputs.

Under risk-neutral inputs, let

$$\Pi(K) = \mathbb{E}_t[(K - B_T)^+]$$

denote the undiscounted basket-put price. For every dependence structure consistent with the supplied marginals,

$$\Pi(K) \leq \Pi^C(K). \quad (65)$$

Importantly, this upper bound can be constructed entirely from constituent put prices. Let q^* satisfy

$$\sum_{i=1}^d w_i Q_{i,t}(q^*) = K, \quad (66)$$

and define the corresponding common-rank strikes $K_i^* = Q_{i,t}(q^*)$. Proposition 2 then gives

$$\Pi^C(K) = \sum_{i=1}^d w_i \Pi_i(K_i^*), \quad \Pi_i(k) := \mathbb{E}_t[(k - S_{i,T})^+]. \quad (67)$$

The same representation has a stronger static-hedging interpretation.

Proposition 7 (Cheapest constituent-put superhedge). *Let $w_i > 0$ and suppose the constituent marginal CDFs at maturity T are continuous and strictly increasing on the relevant strike range. Then*

$$\Pi^C(K) = \min_{\{K_i: \sum_i w_i K_i = K\}} \sum_{i=1}^d w_i \Pi_i(K_i). \quad (68)$$

The minimum is attained at the common-rank strikes $K_i^ = Q_{i,t}(q^*)$, where q^* is defined by (66).*

Proof. See Appendix F. □

Thus the comonotonic upper bound is simultaneously the largest basket-put price over admissible dependence structures and the cheapest static superhedge formed from constituent puts.

A particularly intuitive constituent-option hedge instead protects the same moneyness in every single name. Writing

$$S_{B,t} = \sum_{i=1}^d w_i S_{i,t}, \quad m = \frac{K}{S_{B,t}},$$

choose $K_i^{\text{SM}} = m S_{i,t}$. Since $\sum_i w_i K_i^{\text{SM}} = K$, buying the constituent puts with the portfolio weights guarantees pathwise protection of the same basket strike. Its price is

$$\Pi^{\text{SM}}(K) = \sum_{i=1}^d w_i \Pi_i(m S_{i,t}). \quad (69)$$

Because Proposition 7 minimizes over all such strike allocations,

$$\Pi(K) \leq \Pi^{\text{C}}(K) \leq \Pi^{\text{SM}}(K). \quad (70)$$

The same-moneyness hedge is weakly more expensive because equal moneyness need not correspond to equal marginal rank. The two bounds coincide precisely when the same-moneyness strikes also share a common marginal rank. In particular, this holds throughout the strike range when the constituent marginal laws are identical after normalization by their current levels. The comonotonic construction therefore removes the additional conservatism imposed by treating equal moneyness as equal rank.

This result is particularly useful for margining short basket options. Under a specified shock to the constituent marginals, $\Pi^{\text{C}}(K)$ gives the largest basket-put value consistent with those marginals over all dependence structures. At the same time, (67) expresses this bound entirely through single-name put prices. This is particularly attractive for bespoke baskets, which may have little or no competitive secondary market, while options on their constituents often trade in liquid markets. The comonotonic bound therefore provides a dependence-conservative and market-anchored basis for collateral requirements.

Figure 4 illustrates the magnitude of these bounds. In this calibration, holding the constituent marginals fixed while removing diversification materially changes the downside distribution and basket-put values. The ordering $\Pi(K) \leq \Pi^{\text{C}}(K) \leq \Pi^{\text{SM}}(K)$ holds across strikes: the gap between the fitted and comonotonic prices measures the maximum upward repricing attributable to dependence risk, while the additional gap to the same-moneyness hedge measures the conservatism of imposing equal moneyness rather than equal rank.

12.2 Portfolio comparison beyond variance risk

Variance-based portfolio construction remains a common benchmark in institutional asset allocation. Yet portfolios that are equivalent under a covariance criterion need not have similar distributional or tail risk. Our architecture allows such portfolios to be compared directly through their full portfolio laws.

We construct several fully invested long-only portfolios satisfying a common variance target,

$$w^\top \widehat{\Sigma} w = \sigma_{\text{target}}^2, \quad (71)$$

and evaluate the portfolio law implied by each allocation under the same market state. Figure 5 shows two representative portfolios. Although they carry the same risk by construction under the historical covariance criterion, their portfolio laws differ materially, particularly in the tails.

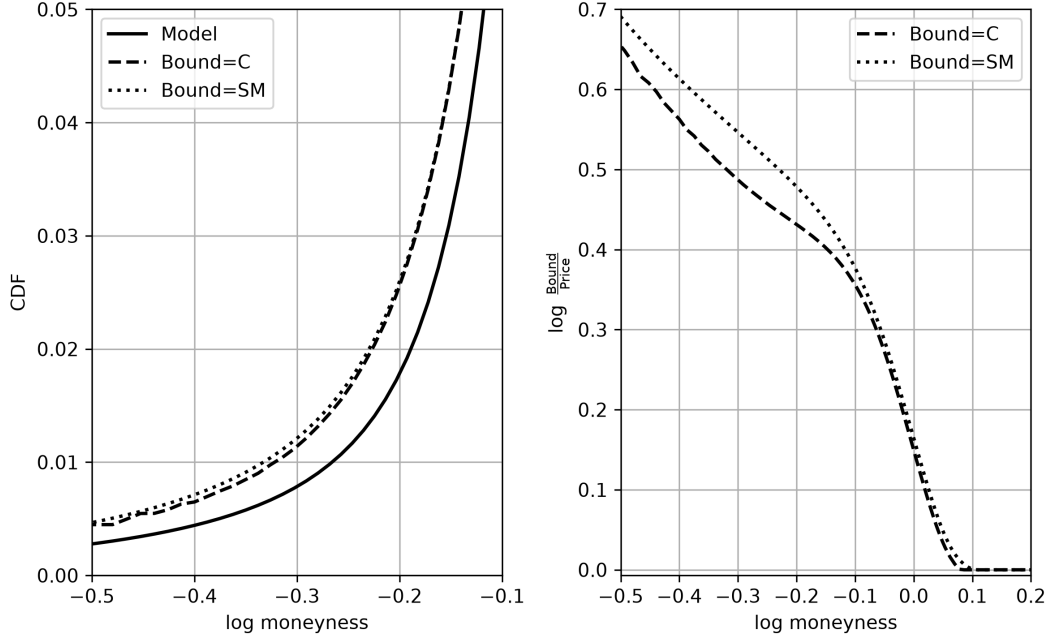


Figure 4: Dependence bounds for a CHF-denominated basket with a 90-day horizon on February 10, 2026. The basket consists of the SMI, SPX, NDX, DAX, FTSE 100, CAC 40, and AEX with weights (0.505, 0.312, 0.020, 0.024, 0.062, 0.068, 0.009). Left: fitted risk-neutral basket CDF, comonotonic CDF (C), and CDF associated with the same-moneyness constituent-put superhedge (SM). Right: log price premia $\log(\Pi^C(K)/\Pi(K))$ and $\log(\Pi^{SM}(K)/\Pi(K))$ relative to the fitted European basket-put price, as functions of basket moneyness $m = K/S_{B,t}$.

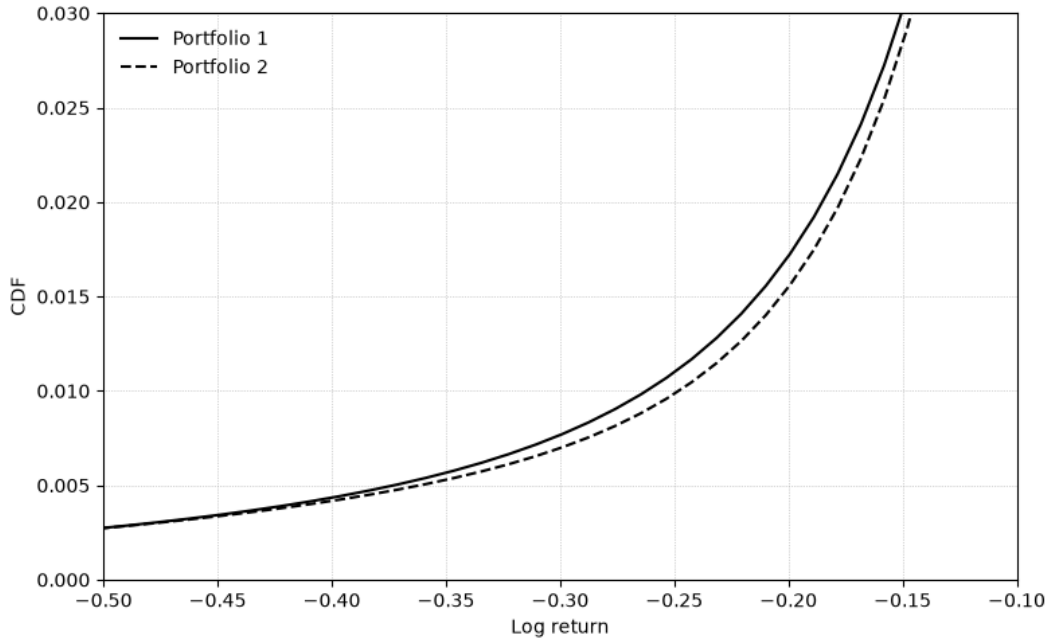


Figure 5: Portfolio laws for two representative equal-variance portfolios. The portfolios are fully invested, long-only, and CHF-denominated, and are constructed on February 10, 2026 from seven major equity indices using a 20-year estimate of the 90-day covariance matrix and a target 90-day volatility of 7%. In the order (SMI, SPX, NDX, DAX, FTSE 100, CAC 40, AEX), their weights are (0.505, 0.312, 0.020, 0.024, 0.062, 0.068, 0.009) and (0.672, 0.125, 0.013, 0.013, 0.022, 0.075, 0.081). Despite the common variance target, the implied portfolio laws differ materially in the tails.

This illustrates the additional information contained in the full portfolio law. Allocations treated as equivalent by a variance criterion can differ materially in economically relevant regions of the distribution. Portfolio construction can therefore assess candidate allocations directly through quantities such as deep downside probabilities or tail losses, rather than through variance alone.

13 Conclusion

This paper develops a rank-space architecture for evaluating the full distribution of high-dimensional portfolios. The key idea is to separate marginal distributions from dependence and exploit the fact that these inputs may evolve at very different frequencies. Marginals enter only through their quantile functions and can therefore be updated as new market, forecasting, or scenario information arrives, while the more expensive dependence layer can be prepared offline and reused. The additive portfolio structure then permits the resulting law to be assembled along a one-dimensional aggregate axis rather than a full high-dimensional state grid. We instantiate this architecture with depth-one Bernstein vines, for which the discretized portfolio law admits exact contractions and polynomial complexity in the portfolio dimension. Conditional on the supplied law inputs, the remaining numerical errors can be controlled *ex ante*, including uniformly controlled errors in portfolio quantiles and tails. The numerical benchmark confirms that these properties remain operational in genuinely high-dimensional settings: the complete portfolio law for a 100-constituent basket can be evaluated in under 0.1 seconds on a single GPU.

The resulting object is the portfolio law itself rather than a particular pricing or risk functional. Under risk-neutral inputs, it yields an arbitrage-consistent pricing law from which basket claims across strikes can be evaluated. Under physical or scenario inputs, the same computational architecture delivers tail probabilities, quantiles, Expected Shortfall, and other distributional risk measures. Separating marginals from dependence also makes dependence risk directly operational. Current marginals can be evaluated under a fitted dependence model, adverse dependence scenarios, or deliberately conservative specifications without changing the computational machinery. For positive portfolio weights, the comonotonic coupling provides a particularly transparent conservative benchmark. Comparing fitted, stressed, and comonotonic portfolio laws quantifies the diversification credit granted by a dependence model and provides a natural basis for hedging, collateral, and margin decisions under current market conditions.

Depth one is the canonical implementation developed in this paper because it keeps the contraction state small and therefore offers the most favorable scalability. It is not a fundamental restriction of the rank-space architecture. At depth one, the dependence topology itself is a design variable: any spanning tree is admissible, and the $d - 1$ modeled pair links can be allocated toward the relationships most relevant for the portfolio. The D-vine path is the lowest-width canonical implementation, while more general trees with moderate hubs provide additional flexibility before the more expensive step of increasing vine depth. For plain Bernstein vines, greater vine depths can likewise be accommodated by enlarging the contraction state. This exposes an explicit trade-off between dependence resolution and computational cost. At a given depth, topology can first be optimized without increasing the number of modeled pair links, while the Bernstein degree controls the resolution of those links. Additional vine depth can then be introduced when the resulting gain in dependence accuracy justifies the larger separator states. This flexibility is particularly valuable in very large portfolios, where dependence capacity can be allocated selectively before more computationally expensive increases in model depth become attractive.

More broadly, predictable access to an entire portfolio distribution opens applications in which distributional evaluation must be repeated inside a larger decision problem. Portfolio optimization

is a natural example: candidate portfolios can be compared using their full laws rather than a small collection of moments or a single risk statistic. The same principle applies to dynamic hedging, stress testing, and margin systems that repeatedly reevaluate exposures as marginals or dependence assumptions change. The broader implication is that richer high-dimensional dependence need not be separated from computationally practical portfolio evaluation. By treating dependence resolution as a controllable computational choice and rapidly changing marginals as online inputs, the architecture provides a scalable route both to pricing and evaluating risk in large portfolios.

A Proofs for Sections 2 and 3

Continuous marginals. For $w_r > 0$, the event $Z_r(U_r) \in [kh, (k+1)h)$ is equivalent to

$$\frac{a_r + kh}{w_r} \leq Q_r(U_r) < \frac{a_r + (k+1)h}{w_r}.$$

Since Q_r is non-decreasing, its preimage is the rank interval with endpoints obtained by applying F_r . For $w_r < 0$, the native-state endpoints reverse; sorting their two CDF values therefore gives the corresponding rank interval in either case.

Atoms and generalized quantiles. Let $F(s_0^-) = \lim_{x \uparrow s_0} F(x)$ and suppose $F(s_0) - F(s_0^-) = a > 0$. By the definition of the generalized quantile,

$$Q(u) = s_0 \quad \text{for } u \in (F(s_0^-), F(s_0)]. \quad (72)$$

If the contribution $w_r s_0 - a_r$ lies in cell $I_{r,k}$, the local factor mass contributed by the atom is therefore

$$\int_{F(s_0^-)}^{F(s_0)} g_{r,\eta}(u) du, \quad (73)$$

not the atom probability multiplied by a factor evaluated at one arbitrary rank. Multiple atoms and continuous intervals are handled by summing the integrals over all components of the preimage.

Proof of Proposition 1. For every coordinate,

$$|Z_r - D_{r,h}(Z_r)| \leq \frac{h}{2}.$$

The possibly padded part of the final cell carries zero mass and therefore does not affect this bound. Hence, by the triangle inequality,

$$|B_T - B_{T,h}| \leq \sum_{r=1}^d |Z_r - D_{r,h}(Z_r)| \leq \frac{dh}{2} = \varepsilon_h,$$

which proves (i).

For (ii), Lipschitz continuity gives pointwise

$$|\Phi(B_T) - \Phi(B_{T,h})| \leq L_\Phi |B_T - B_{T,h}| \leq L_\Phi \varepsilon_h.$$

Taking expectations yields the stated bound.

For (iii), the pathwise inequality implies

$$\{B_T \leq x - \varepsilon_h\} \subseteq \{B_{T,h} \leq x\} \subseteq \{B_T \leq x + \varepsilon_h\}.$$

Taking probabilities gives

$$F_B(x - \varepsilon_h) \leq F_{B_h}(x) \leq F_B(x + \varepsilon_h).$$

For (iv), if the density of B_T is bounded by $\bar{f}_B$, then

$$F_B(x + \varepsilon_h) - F_B(x) \leq \bar{f}_B \varepsilon_h, \quad F_B(x) - F_B(x - \varepsilon_h) \leq \bar{f}_B \varepsilon_h,$$

and the result follows from (iii).

Finally, (i) implies

$$B_{T,h} \leq B_T + \varepsilon_h, \quad B_T \leq B_{T,h} + \varepsilon_h \quad \text{a.s.}$$

Generalized quantiles preserve almost-sure stochastic ordering, and translation shifts quantiles by the same constant. Therefore, for every $p \in (0, 1)$,

$$Q_{B_h}(p) \leq Q_B(p) + \varepsilon_h, \quad Q_B(p) \leq Q_{B_h}(p) + \varepsilon_h.$$

Thus

$$|Q_B(p) - Q_{B_h}(p)| \leq \varepsilon_h$$

uniformly in p , which proves (v). $\square$

Proof of Proposition 2. (i) All three shape properties are properties of $K \mapsto \mathbb{E}[(K - B)^+]$ for a fixed integrable B : put–call parity is the identity $(K - B)^+ - (B - K)^+ = K - B$; monotonicity and convexity follow because $K \mapsto (K - B)^+$ is nondecreasing and convex pointwise in B ; and Jensen’s inequality gives $\Pi(K) \geq (K - \mathbb{E}_t[B_T])_+$. The marginals are matched by construction: if U_r is uniform on $(0, 1)$, then $Q_{r,t}(U_r)$ has distribution $F_{r,t}$ by the generalized-quantile construction of Theorem 1. Hence every constituent’s vanilla options are repriced exactly.

(ii) For fixed marginals, the sum is bounded in convex order below by its mean (Jensen) and above by the comonotonic sum (Dhaene et al., 2002b), which gives (18).

(iii) Comonotonicity makes quantiles additive, $Q_{BC}(v) = \sum_r w_r Q_{r,t}(v)$ for $w_r > 0$. With $v^* := F_{BC}(K)$ and $K_r := Q_{r,t}(v^*)$, the event $\{B_T^C \leq K\}$ coincides with $\{V \leq v^*\} = \bigcap_r \{S_{r,T} \leq K_r\}$, and on that event $(K - B_T^C)^+ = \sum_r w_r (K_r - S_{r,T})^+$ coordinatewise (up to the usual convention at flat parts of the quantile functions). Taking expectations gives (19); see also Dhaene et al. (2002a); Deelstra et al. (2004). $\square$

B Proofs for Section 4

Proof of Proposition 3. Nonnegativity of (20) is clear. We prove total mass one by induction on d . For $d = 1$ the edge set is empty and the product is the constant 1. For $d \geq 2$, the tree $\mathcal{G}$ has a leaf ℓ with a unique neighbor $\nu(\ell)$. The only factor in (20) involving u_ℓ is $c_{(\ell, \nu(\ell))}(u_\ell, u_{\nu(\ell)})$, and since every c_e has uniform margins,

$$\int_0^1 c_{(\ell, \nu(\ell))}(u_\ell, u_{\nu(\ell)}) du_\ell = 1 \quad \text{for every } u_{\nu(\ell)} \in [0, 1].$$

Integrating out u_ℓ therefore removes exactly that factor and leaves the first-tree density of the subtree on the remaining $d - 1$ vertices, which integrates to one by the induction hypothesis.

It remains to show that every one-dimensional margin is uniform. Fix a coordinate r and integrate out all other coordinates. As long as at least two vertices remain, the current tree has a leaf different from r ; integrating out that leaf removes its incident factor exactly as above and leaves a first-tree density on one fewer vertex. Repeating the peeling until only r remains removes all $d - 1$ factors, so the marginal density of u_r is the constant 1 on $[0, 1]$. $\square$

Proof of Proposition 4. (i) follows from the beta-integral identity. For (ii), the Bernstein derivative formula

$$\frac{d}{du} b_{j,n+1}(u) = (n+1) \{b_{j-1,n}(u) - b_{j,n}(u)\}$$

implies

$$\frac{d}{du} \left[\frac{1}{n+1} \sum_{j=k+1}^{n+1} b_{j,n+1}(u) \right] = b_{k,n}(u)$$

by telescoping; the bracketed expression vanishes at $u = 0$, which proves the integrated-basis identity. For (iii), direct multiplication of the Bernstein basis functions gives

$$b_{i,n}(u)b_{j,m}(u) = \frac{\binom{n}{i}\binom{m}{j}}{\binom{n+m}{i+j}} b_{i+j,n+m}(u).$$

Finally, (iv) follows from Vandermonde's identity, $\sum_i \binom{n}{i} \binom{m}{\ell-i} = \binom{n+m}{\ell}$, together with nonnegativity of the binomial coefficients. The first partial moment and second integrated basis used later are recorded in Appendix G. $\square$

C Exact Contractions for C-Vine

C.1 The C-vine as a normalized root recursion

Let the observed root be indexed by 0 for notational simplicity and let $m = d - 1$ be the number of leaves. For leaf i , write the pair density as

$$c_i(u_0, u_i) = \sum_{a=0}^{n_c} b_{a,n_c}(u_0) g_{i,a}(u_i), \quad g_{i,a}(u) := \sum_{b=0}^{n_c} A_{ab}^{(i)} b_{b,n_c}(u). \quad (74)$$

Multiplying the m pair factors along the root coordinate generates Bernstein degrees $n_c, 2n_c, \dots, mn_c$. The unstable ordinary-polynomial approach would absorb $\binom{n_c}{a}$ into every $g_{i,a}$ and divide the root state by $\binom{mn_c}{A}$; those factors have exponential dynamic range in m . We instead remain in normalized Bernstein coordinates throughout.

Set $H_a^{(1)} = g_{1,a}$. For $j = 1, \dots, m - 1$, define

$$H_\ell^{(j+1)} = \sum_{k+a=\ell} H_k^{(j)} g_{j+1,a} \omega_{jn_c, n_c}(k, a), \quad \ell = 0, \dots, (j+1)n_c. \quad (75)$$

Theorem 4 (Normalized C-vine root aggregation). *For the plain Bernstein basis,*

$$c_C(u) = \sum_{A=0}^{mn_c} b_{A,mn_c}(u_0) H_A^{(m)}(u_1, \dots, u_m), \quad (76)$$

where $H^{(m)}$ is generated by (75). Every recursion weight is nonnegative, no larger than one, and the weights contributing to a fixed output index sum to one. The root-state size is

$$N_\star = mn_c, \quad R_\star = N_\star + 1. \quad (77)$$

Proof. Induction on the leaves via the Bernstein product identity in Proposition 4(iii); see Appendix D. $\square$

C.2 C-vine: rank-cell arrays and aggregate-axis recursion

For leaf i , define

$$Q_{i,a,k} := \int_{u_{i,k}^-}^{u_{i,k}^+} g_{i,a}(u) du = \sum_{b=0}^{n_c} A_{ab}^{(i)} \Delta B_{b,n_c}[u_{i,k}^-, u_{i,k}^+] \quad (78)$$

in the plain basis. The root arrays are

$$R_{A,k} := \Delta B_{A,N_*}[u_{0,k}^-, u_{0,k}^+]. \quad (79)$$

The coefficient recursion and the aggregate-cell convolution are performed simultaneously. Initialize $H_a^{(1)} = Q_{1,a}$. Then

$$H_\ell^{(j+1)} = \sum_{k+a=\ell} \omega_{jn_c, n_c}(k, a) \left(H_k^{(j)} * Q_{j+1,a} \right). \quad (80)$$

Finally,

$$\pi^{(h)} = \sum_{A=0}^{N_*} H_A^{(m)} * R_A. \quad (81)$$

Equations (80)–(81) are exact for the quantized portfolio law. In Fourier space, every aggregate-axis convolution becomes a pointwise product, while the normalized coefficient-axis recursion remains explicit.

Algorithm 3: Full plain-basis C-vine build

1. Construct the common finite lattice and every coordinate's rank-cell endpoints.
2. For each leaf, assemble $Q_{i,a}$ from (78), pad to length L , and transform along the aggregate axis.
3. Starting from the first leaf, apply the normalized recurrence (80) frequency by frequency. Do not multiply leaves by $\binom{n_c}{a}$ and do not divide the root by $\binom{N_*}{A}$.
4. Assemble and transform the R_* root arrays (79).
5. Close the root state with (81), invert the aggregate-axis transform, and run mass, sign, and payoff-shape diagnostics.

D Proofs for Section 5

Proof of Theorem 3. Expand the matrix product over bond indices $j_1, \dots, j_{d-1} \in \{0, \dots, n_c\}$. The right-hand side of (31) equals

$$\sum_{j_1, \dots, j_{d-1}} \left[b(u_1)^\top A^{(1)} \right]_{j_1} \prod_{r=2}^{d-1} b_{j_{r-1}, n_c}(u_r) \left[b(u_r)^\top A^{(r)} \right]_{j_r} b_{j_{d-1}, n_c}(u_d).$$

Regrouping the factors coordinate by coordinate, the summand is the product over $r = 1, \dots, d-1$ of the edge factors: at edge $(r, r+1)$, the indices $a = j_{r-1}$ (from the previous bond) and $b = j_r$ (from the next) select

$$\sum_{a,b} A_{ab}^{(r)} b_{a,n_c}(u_r) b_{b,n_c}(u_{r+1}) = c_r(u_r, u_{r+1}),$$

with the convention that at $r = 1$ the factor $[b(u_1)^\top A^{(1)}]_{j_1} = \sum_a A_{aj_1}^{(1)} b_{a,n_c}(u_1)$ supplies the $b_{j_1, n_c}(u_2)$ index implicitly through the next core. The summand is thus exactly the product of pair densities in (21), and summing over all bond indices gives the identity. $\square$

Proof of Theorem 4. For $j = 1$, (76) is the first pair factor (74). Suppose it holds after j leaves: $c^{(j)}(u) = \sum_{k=0}^{jn_c} b_{k,jn_c}(u_0) H_k^{(j)}(u_1, \dots, u_j)$. Multiplying by the $(j+1)$ -st pair factor $\sum_{a=0}^{n_c} b_{a,n_c}(u_0) g_{j+1,a}(u_{j+1})$ and applying Proposition 4(iii) termwise gives

$$b_{k,jn_c}(u_0) b_{a,n_c}(u_0) = \omega_{jn_c,n_c}(k, a) b_{k+a,(j+1)n_c}(u_0).$$

Collecting all terms with $k + a = \ell$ yields (75), and the degree range $\ell \in \{0, \dots, (j+1)n_c\}$ is exact. The weights are nonnegative and bounded by one, and for fixed ℓ they sum to one, by Proposition 4(iv). The root-state size follows since the maximal root degree after m leaves is mn_c . $\square$

E Proofs for Section 7

Proof of Proposition 6. At the left endpoint, $\Lambda_\nu(0) = 0$ and $b_{a,n_c}(0) = \delta_{a0}$, so $\varphi_a(0) = \delta_{a0}/\nu_0$ by (50); similarly $\varphi_a(1) = \delta_{a,n_c}/\nu_{n_c}$. Hence

$$c_e(0, 0) = \frac{A_{00}^{(e)}}{\nu_0^2}, \quad c_e(1, 1) = \frac{A_{n_c,n_c}^{(e)}}{\nu_{n_c}^2}, \quad c_e(0, 1) = \frac{A_{0,n_c}^{(e)}}{\nu_0 \nu_{n_c}}.$$

The row constraint in (52) gives $A_{00} \leq (n_c + 1)\nu_0$, hence $c_e(0, 0) \leq (n_c + 1)/\nu_0$; analogously $c_e(1, 1) \leq (n_c + 1)/\nu_{n_c}$. For the mixed corners, the row and column constraints together give $A_{0,n_c} \leq (n_c + 1) \min\{\nu_0, \nu_{n_c}\}$, which yields (54). Since the density is continuous and bounded on $[0, 1]^2$, $\Pr(U \leq q, V \leq q) = \int_0^q \int_0^q c_e(u, v) du dv$ is $O(q^2)$ as $q \downarrow 0$, and likewise for the upper-right corner. $\square$

F Proofs for Section 12

Proof of Proposition 7. For any strikes $(K_i)_{i=1}^d$ satisfying $\sum_i w_i K_i = K$, convexity of the positive-part operator gives

$$\left(K - \sum_i w_i S_{i,T} \right)^+ \leq \sum_i w_i (K_i - S_{i,T})^+.$$

Taking expectations under the comonotonic coupling yields

$$\Pi^C(K) \leq \sum_i w_i \Pi_i(K_i).$$

Now let q^* satisfy

$$\sum_i w_i Q_{i,t}(q^*) = K$$

and set $K_i^* = Q_{i,t}(q^*)$. Under the comonotonic coupling, all constituent puts are simultaneously in the money exactly when $V \leq q^*$, so

$$(K - B_T^C)^+ = \sum_i w_i (K_i^* - S_{i,T})^+$$

pathwise. Hence

$$\Pi^C(K) = \sum_i w_i \Pi_i(K_i^*),$$

which attains the lower bound above and proves the result. $\square$

G Detailed Bernstein Rank-Cell Formulas

G.1 General products at one node

For indices $a_1, \dots, a_m \in \{0, \dots, n_c\}$ and $A = \sum_j a_j$, repeated use of Proposition 4(iii) gives

$$\prod_{j=1}^m b_{a_j, n_c}(u) = \frac{\prod_{j=1}^m \binom{n_c}{a_j}}{\binom{mn_c}{A}} b_{A, mn_c}(u). \quad (82)$$

Therefore its exact mass on rank interval $[u^-, u^+]$ is the displayed scalar times $\Delta B_{A, mn_c}[u^-, u^+]$. This is the local formula for a generic first-tree node before coefficient matrices are contracted.

G.2 D-vine interior core

Starting from

$$G_r(u)_{ij} = b_{i, n_c}(u) \sum_{a=0}^{n_c} A_{aj}^{(r)} b_{a, n_c}(u),$$

linearity and the two-factor product identity give

$$\int_{u^-}^{u^+} G_r(u)_{ij} du = \sum_a A_{aj}^{(r)} \frac{\binom{n_c}{i} \binom{n_c}{a}}{\binom{2n_c}{i+a}} \Delta B_{i+a, 2n_c}[u^-, u^+],$$

which is (36).

G.3 C-vine normalization

The unstable scalar in (82) depends on the full multi-index, not only on A . The normalized recursion avoids separating it into exponentially large leaf factors and an exponentially small root factor. At each multiplication step, Vandermonde's identity writes the degree- $(j+1)n_c$ coefficient as a convex combination of products of degree- jn_c and degree- n_c coefficients. This proves both the algebraic identity and the bound in Theorem 4.

G.4 Partial moments of the basis

The first partial moment and twice-integrated basis are

$$M_{k,n}(u) := \int_0^u v b_{k,n}(v) dv = \frac{k+1}{(n+1)(n+2)} \sum_{j=k+2}^{n+2} b_{j,n+2}(u), \quad (83)$$

$$\mathbb{B}_{k,n}(u) := \int_0^u B_{k,n}(v) dv = \frac{1}{(n+1)(n+2)} \sum_{j=k+2}^{n+2} (j-k-1) b_{j,n+2}(u). \quad (84)$$

Both identities follow from Proposition 4(ii) and $v b_{k,n}(v) = \frac{k+1}{n+2} b_{k+1, n+1}(v)$. They are recorded for completeness.

H Cache Derivations, Memory Accounting, and Scaling

H.1 D-vine prefixes and suffixes

At one Fourier frequency, define

$$\vec{V}_1 = \hat{L}_1^\top, \quad \vec{V}_r = \vec{V}_{r-1} \hat{G}_r, \quad 2 \leq r \leq d-1, \quad (85)$$

$$\overleftarrow{V}_d = \hat{R}_d, \quad \overleftarrow{V}_r = \hat{G}_r \overleftarrow{V}_{r+1}, \quad 2 \leq r \leq d-1. \quad (86)$$

A baseline replacement of core r uses $\vec{V}_{r-1}$ and $\overleftarrow{V}_{r+1}$ once. If the core is accepted, $\vec{V}_j$ changes for every $j \geq r$ and $\overleftarrow{V}_j$ changes for every $j \leq r$. Rebuilding all affected vectors requires a linear number of vector-matrix products and hence $O(dR^2L)$ work.

Streaming the full build needs a current R -vector and one $R \times R$ core per frequency, or $O(R^2L)$ memory. Retaining every transformed local core costs $O(dR^2L)$; retaining all prefix and suffix vectors adds $O(dRL)$. These quantities must not be conflated.

H.2 C-vine exclusions

Let $\mathcal{P}_i$ denote the aggregate of all leaf polynomials except leaf i , represented in normalized Bernstein coefficient coordinates. A baseline leaf bump combines the new degree- n_c leaf with $\mathcal{P}_i$ and then closes the root. If the bump is accepted, $\mathcal{P}_j$ changes for every $j \neq i$. Storing all exclusions costs $O(dR_*L)$ in addition to the $O(dRL)$ leaf slabs. A reference implementation that stores only the full aggregate uses much less memory but must rebuild that aggregate after every accepted leaf update.

H.3 Final-output cache

After the inverse transform, retaining mass, CDF, and integrated-put arrays costs $O(L)$ memory. On a uniform grid, a strike query then uses constant-time indexing and interpolation. Without this output cache, calling a CDF or put routine may repeat the inverse transform and cumulative pass; preprocessing and output reconstruction must therefore be included in runtime tables.

H.4 Warped-root scaling and stability checks

A simple statewise scaling for the warped C-vine root of Section 7.4 is

$$s_A := \max\{\|\mathbf{R}_A\|_1, \epsilon\}, \quad \tilde{\mathbf{R}}_A = \mathbf{R}_A/s_A, \quad \tilde{\mathbf{H}}_A = s_A \mathbf{H}_A, \quad (87)$$

which leaves (81) unchanged. Root states should record $m \log \Lambda'_\nu(u)$ before exponentiation. Small interval masses should be evaluated with a stable beta-CDF interval routine or a local recurrence rather than by subtracting nearby CDF values. Dynamic-range tests must accompany any claim of stability as d grows: report the ratio (57), the conditioning of the root-state norms $\|\mathbf{R}_A\|_1$ across A , and the signed FFT residual before any projection.

I Empirical Construction of the Dependence Layer

This appendix describes one empirical construction of the vine topology and Bernstein coefficient matrices treated as fixed offline inputs in the main text. We now turn to the empirical construction of this dependence layer. The objective is not to require the Bernstein vine itself to be the preferred

statistical model of dependence. Rather, it provides the contractible representation into which a flexible dependence model can be compiled for subsequent fast evaluation of the portfolio law.

For a physical-measure implementation, let

$$U_t = (U_{1t}, \dots, U_{dt}), \quad t = 1, \dots, N,$$

denote pseudo-observations obtained after filtering the marginal dynamics of the constituents (Genest et al., 1995; Joe, 2014). Dependence estimation then takes place entirely in rank space and remains separate from the marginal models used in subsequent portfolio evaluation. Under risk-neutral or scenario inputs, the same construction applies with a dependence model calibrated or otherwise specified under the corresponding measure. The computational architecture itself does not require the dependence specification to be the same across measures.

For a Bernstein pair copula of degree n_c , recall that $R = n_c + 1$. A direct estimator can be constructed from empirical cell probabilities on the $R \times R$ rank grid. Finite-sample cell counts generally fail to satisfy the required copula margins, so we rescale the cell matrix by iterative proportional fitting while preserving nonnegativity. For warped pair copulas, the corresponding margins are determined by the fixed warp. Increasing the degree refines the dependence resolution at the quadratic computational cost in R characterized in Corollaries 1 and 2.

Fine direct estimation, however, quickly becomes data intensive. Under independence, a corner cell has probability R^{-2} , giving its empirical probability a relative standard error of approximately $R/\sqrt{N}$. Relative precision ϵ therefore requires roughly

$$N \gtrsim \frac{R^2}{\epsilon^2}. \quad (88)$$

At $R = 20$ and $\epsilon = 0.1$, this corresponds to about 4×10^4 effectively independent observations, compared with roughly 5×10^3 daily observations over twenty years before accounting for serial dependence (Nagler et al., 2022; Oh and Patton, 2018). Positive tail dependence makes the corner cells less sparse, but the calculation illustrates why finely resolved Bernstein copulas can be difficult to estimate directly from historical data.

We therefore separate statistical estimation from computational representation. First, we fit a flexible *teacher copula* to the historical pseudo-observations. The teacher should reflect the dependence model one would ideally choose for the data, including appropriate interpolation and extrapolation in sparsely observed regions, without imposing tractability requirements arising from repeated portfolio evaluation. It may therefore be any suitable copula specification, even if computing high-dimensional portfolio distributions under it is expensive.

The vine topology is selected before fitting the student coefficients. For a general first tree, standard selection amounts to a maximum-spanning-tree problem using an edge score such as $|\tau_{rs}|$ (Dißmann et al., 2013). For portfolio applications, it is natural to account also for the economic importance of the constituents. We therefore consider the portfolio-weighted score

$$s_{rs} = |w_r| \sigma_r |w_s| \sigma_s |\tau_{rs}|, \quad (89)$$

where σ_r is a scale measure for constituent r at the relevant horizon. The unrestricted portfolio-aware first tree is obtained by using s_{rs} as the edge weight in maximum-spanning-tree selection.

D- and C-vines arise as constrained versions of the same problem. For a D-vine, the path restriction leads to a weighted ordering problem that favors placing strongly connected and economically important constituents next to one another. For a C-vine, node-level sums of the edge scores provide a natural criterion for selecting the root. More generally, maximum node degree or other

structural constraints can be imposed when the desired dependence topology must also respect a computational budget.

The shallow Bernstein vine is then trained offline to reproduce the fitted teacher. For the plain Bernstein student, partition each rank coordinate into $J_a = [a/R, (a+1)/R)$, $a = 0, \dots, R-1$, with the last interval closed at one. Conditional on the selected first-tree topology, draw $U^{(1)}, \dots, U^{(N_{\text{syn}})}$ independently from the teacher. For each edge $e = (r, s)$, define the synthetic cell probabilities

$$\hat{P}_{ab}^{(e)} = \frac{1}{N_{\text{syn}}} \sum_{m=1}^{N_{\text{syn}}} \mathbb{1}_{\{U_r^{(m)} \in J_a, U_s^{(m)} \in J_b\}}.$$

We project $\hat{P}^{(e)}$ by iterative proportional fitting, equivalently alternating row-column scaling, onto the set of nonnegative matrices with row and column sums $1/R$ (Sinkhorn and Knopp, 1967). With the positive pseudocount used above, $\hat{P}^{(e)}$ is strictly positive, so the Sinkhorn–Knopp iteration converges to the required uniform-margin scaling. If the pseudocount is omitted and structural zeros are allowed, the corresponding support conditions for convergence must be verified (Sinkhorn and Knopp, 1967). Denoting the result by $\tilde{P}^{(e)}$, the Bernstein coefficients are

$$A_{ab}^{(e)} = R^2 \tilde{P}_{ab}^{(e)}. \quad (90)$$

They therefore satisfy the copula constraints in (25) exactly. For a fixed warped basis, the corresponding IPF targets are given by its prescribed marginal weights. Because this step is performed only when the dependence layer is revised, the teacher can be sampled sufficiently extensively to estimate the Bernstein coefficients with high precision. These synthetic draws add no statistical information beyond the original data; their role is to transfer the fitted teacher into the contractible representation used for fast online evaluation.

The teacher is first evaluated against held-out empirical pseudo-observations. Conditional on the accepted teacher, the teacher–student approximation is assessed on an independent synthetic validation sample. Teacher and student are compared in terms of pairwise dependence, finite-level lower- and upper-tail probabilities and, most importantly for our application, the aggregate portfolio distributions they imply. The Bernstein degree is increased until these representation discrepancies fall below pre-specified tolerances. Where the dependence layer is intended for reuse across several portfolios, this comparison can be performed over a representative set of portfolio weights and marginal configurations. The finite-level tail properties associated with the Bernstein resolution are discussed further in Section 7.

Teacher estimation, student training, degree selection, and topology selection all belong to the offline stage and may therefore be computationally intensive without affecting subsequent portfolio evaluation. Once the fitted dependence layer has been compiled into the matrices $A^{(e)}$, the online architecture developed above can incorporate changing marginal distributions without repeating this estimation procedure.

Error budget. The different approximation layers should be kept conceptually separate. Dependence-model and representation choices are determined offline and assessed empirically, whereas the numerical approximation of the resulting portfolio law is controlled conditional on those inputs. Table 3 summarizes this distinction.

Source	Type	Control or diagnostic
Teacher dependence model	statistical / model	held-out empirical diagnostics and comparison across plausible dependence specifications
Vine topology and depth	representation / structural	offline model selection and comparison across dependence specifications
Teacher–student Bernstein representation	representation	held-out teacher–student discrepancies in dependence measures, finite-level tails, and portfolio laws; increase n_c as required
Portfolio lattice h	numerical	deterministic bound $\varepsilon_h = dh/2$ from Proposition 1
Tail truncation or clipping	numerical / input truncation	first-moment bound in Proposition 5
Warped rank-cell integration	numerical	one-dimensional quadrature tolerance; absent for exact plain-Bernstein rank-cell masses
FFT and floating-point arithmetic	implementation	mass normalization, nonnegativity, and numerical-precision diagnostics

Table 3: Sources of approximation and numerical error. Deterministic numerical guarantees are conditional on the supplied marginal laws and the fitted dependence representation.

Conflict of Interest Statement

The authors declare no conflicts of interest.

References

- Aas, K., Czado, C., Frigessi, A., and Bakken, H. (2009). Pair-copula constructions of multiple dependence. *Insurance: Mathematics and Economics*, 44:182–198.
- Ang, A. and Chen, J. (2002). Asymmetric correlations of equity portfolios. *Journal of Financial Economics*, 63(3):443–494.
- Arenstein, L. and Kastoryano, M. (2026). COS–TT–CHF: A tensor-train characteristic-function COS method for multi-asset option pricing. arXiv preprint arXiv:2608.17636.
- Bartl, D., Drapeau, S., and Tangpi, L. (2020). Computational aspects of robust optimized certainty equivalents and option pricing. *Mathematical Finance*, 30:287–309.
- Bayer, C., Ben Hammouda, C., Papapantoleon, A., Samet, M., and Tempone, R. (2023). Optimal damping with a hierarchical adaptive quadrature for efficient Fourier pricing of multi-asset options in Lévy models. *Journal of Computational Finance*, 27(3):43–86.
- Bayer, C., Ben Hammouda, C., Papapantoleon, A., Samet, M., and Tempone, R. (2026). Quasi-monte carlo with domain transformation for efficient Fourier pricing of multi-asset options. arXiv:2403.02832v5. June 2026.
- Becker, S., Cheridito, P., and Jentzen, A. (2019). Deep optimal stopping. *Journal of Machine Learning Research*, 20:1–25.
- Bedford, T. and Cooke, R. M. (2001). Probability density decomposition for conditionally dependent random variables modeled by vines. *Annals of Mathematics and Artificial Intelligence*, 32:245–268.

- Bedford, T. and Cooke, R. M. (2002). Vines: A new graphical model for dependent random variables. *The Annals of Statistics*, 30:1031–1068.
- Bernard, C., Rüschendorf, L., and Vanduffel, S. (2017). Value-at-Risk bounds with variance constraints. *Journal of Risk and Insurance*, 84:923–959.
- Brechmann, E. C., Czado, C., and Aas, K. (2012). Truncated regular vines in high dimensions with application to financial data. *Canadian Journal of Statistics*, 40:68–85.
- Caldana, R. and Fusai, G. (2013). A general closed-form spread option pricing formula. *Journal of Banking & Finance*, 37:4893–4906.
- Caldana, R., Fusai, G., Gnoatto, A., and Grasselli, M. (2016). General closed-form basket option pricing bounds. *Quantitative Finance*, 16:535–554.
- Carr, P. and Madan, D. B. (1999). Option valuation using the fast Fourier transform. *Journal of Computational Finance*, 2:61–73.
- Cottin, C. and Pfeifer, D. (2014). From Bernstein polynomials to Bernstein copulas. *Journal of Applied Functional Analysis*, 9:277–288.
- Czado, C. (2019). *Analyzing Dependent Data with Vine Copulas*. Lecture Notes in Statistics 222. Springer, Cham.
- Deelstra, G., Liinev, J., and Vanmaele, M. (2004). Pricing of arithmetic basket options by conditioning. *Insurance: Mathematics and Economics*, 34:55–77.
- Demarta, S. and McNeil, A. J. (2005). The t copula and related copulas. *International Statistical Review*, 73:111–129.
- Dembo, A. and Zeitouni, O. (1998). *Large Deviations Techniques and Applications*, volume 38 of *Applications of Mathematics*. Springer, New York, 2nd edition.
- Dhaene, J., Denuit, M., Goovaerts, M. J., Kaas, R., and Vyncke, D. (2002a). The concept of comonotonicity in actuarial science and finance: Applications. *Insurance: Mathematics and Economics*, 31:133–161.
- Dhaene, J., Denuit, M., Goovaerts, M. J., Kaas, R., and Vyncke, D. (2002b). The concept of comonotonicity in actuarial science and finance: Theory. *Insurance: Mathematics and Economics*, 31:3–33.
- Dißmann, J., Brechmann, E. C., Czado, C., and Kurowicka, D. (2013). Selecting and estimating regular vine copulae and application to financial returns. *Computational Statistics & Data Analysis*, 59:52–69.
- Eckstein, S., Guo, G., Lim, T., and Obłój, J. (2021). Robust pricing and hedging of options on multiple assets and its numerics. *SIAM Journal on Financial Mathematics*, 12:158–188.
- Embrechts, P., Hofert, M., and Wang, R. (2016). Bernoulli and tail-dependence compatibility. *Annals of Applied Probability*, 26:1636–1658.
- Embrechts, P., McNeil, A. J., and Straumann, D. (2002). Correlation and dependence in risk management: properties and pitfalls. In Dempster, M. A. H., editor, *Risk Management: Value at Risk and Beyond*, pages 176–223. Cambridge University Press, Cambridge.

- Embrechts, P., Puccetti, G., and Rüschendorf, L. (2013). Model uncertainty and VaR aggregation. *Journal of Banking & Finance*, 37:2750–2764.
- Embrechts, P., Wang, B., and Wang, R. (2015). Aggregation-robustness and model uncertainty of regulatory risk measures. *Finance and Stochastics*, 19:763–790.
- Fang, F. and Oosterlee, C. W. (2008). A novel pricing method for European options based on Fourier-cosine series expansions. *SIAM Journal on Scientific Computing*, 31(2):826–848.
- Gaß, M., Glau, K., Mahlstedt, M., and Mair, M. (2018). Chebyshev interpolation for parametric option pricing. *Finance and Stochastics*, 22:701–731.
- Genest, C., Ghouli, K., and Rivest, L.-P. (1995). A semiparametric estimation procedure of dependence parameters in multivariate families of distributions. *Biometrika*, 82:543–552.
- Glasserman, P. (2004). *Monte Carlo Methods in Financial Engineering*, volume 53 of *Stochastic Modelling and Applied Probability*. Springer, New York.
- Glau, K., Kressner, D., and Statti, F. (2020). Low-rank tensor approximation for Chebyshev interpolation in parametric option pricing. *SIAM Journal on Financial Mathematics*, 11:897–927.
- Guo, G. and Oblój, J. (2019). Computational methods for martingale optimal transport problems. *Annals of Applied Probability*, 29:3311–3347.
- Han, J., Jentzen, A., and E, W. (2018). Solving high-dimensional partial differential equations using deep learning. *Proceedings of the National Academy of Sciences*, 115:8505–8510.
- Hobson, D., Laurence, P., and Wang, T.-H. (2005). Static-arbitrage upper bounds for the prices of basket options. *Quantitative Finance*, 5:329–342.
- Hurd, T. R. and Zhou, Z. (2010). A Fourier transform method for spread option pricing. *SIAM Journal on Financial Mathematics*, 1:142–157.
- Joe, H. (2014). *Dependence Modeling with Copulas*. CRC Press, Boca Raton.
- Ju, N. (2002). Pricing Asian and basket options via Taylor expansion. *Journal of Computational Finance*, 5:79–103.
- Junike, G. and Stier, H. (2025). From characteristic functions to multivariate distribution functions and European option prices by the (damped) COS method. *SIAM Journal on Numerical Analysis*, 63(6):2421–2453.
- Kastoryano, M. and Pancotti, N. (2022). A highly efficient tensor network algorithm for multi-asset Fourier options pricing. arXiv:2203.02804.
- Levy, E. (1992). Pricing European average rate currency options. *Journal of International Money and Finance*, 11:474–491.
- Linders, D. and Stassen, B. (2016). The multivariate Variance Gamma model: basket option pricing and calibration. *Quantitative Finance*, 16:555–572.
- Longin, F. and Solnik, B. (2001). Extreme correlation of international equity markets. *The Journal of Finance*, 56(2):649–676.

- McNeil, A. J., Frey, R., and Embrechts, P. (2015). *Quantitative Risk Management: Concepts, Techniques and Tools*. Princeton University Press, Princeton, revised edition.
- Nagler, T., Krüger, D., and Min, A. (2022). Stationary vine copula models for multivariate time series. *Journal of Econometrics*, 227:305–324.
- Neufeld, A., Papapantoleon, A., and Xiang, Q. (2023). Model-free bounds for multi-asset options using option-implied information and their exact computation. *Management Science*, 69:2051–2068.
- Oh, D. H. and Patton, A. J. (2018). Time-varying systemic risk: evidence from a dynamic copula model of CDS spreads. *Journal of Business & Economic Statistics*, 36:181–195.
- Oseledets, I. V. (2011). Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33:2295–2317.
- Pagès, G. and Printems, J. (2003). Optimal quadratic quantization for numerics: the Gaussian case. *Monte Carlo Methods and Applications*, 9(2):135–165.
- Puccetti, G. and Wang, R. (2015). Extremal dependence concepts. *Statistical Science*, 30:485–517.
- Sancetta, A. and Satchell, S. (2004). The Bernstein copula and its applications to modeling and approximations of multivariate distributions. *Econometric Theory*, 20:535–562.
- Scheffer, M. and Weiß, G. N. F. (2017). Smooth nonparametric Bernstein vine copulas. *Quantitative Finance*, 17:139–156.
- Sinkhorn, R. and Knopp, P. (1967). Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348.
- Sklar, A. (1959). Fonctions de répartition à n dimensions et leurs marges. *Publications de l’Institut de Statistique de l’Université de Paris*, 8:229–231.
- Tankov, P. (2011). Improved Fréchet bounds and model-free pricing of multi-asset options. *Journal of Applied Probability*, 48:389–403.