On Constructing a Low-Rank Approximation of a Density

Joshua Gehlen*, Daniel Frisch†, Felix Govaers*, Uwe D. Hanebeck†, Wolfgang Koch*
*Sensor Data and Information Fusion, Fraunhofer FKIE, Wachtberg, Germany,
{joshua.gehlen, felix.govaers, wolfgang.koch}@fkie.fraunhofer.de
†Intelligent Sensor-Actuator-Systems Laboratory (ISAS), Karlsruhe Institute of Technology (KIT), Germany,
{daniel.frisch, uwe.hanebeck}@kit.edu

***Abstract*—Low-rank tensor representations can alleviate the exponential storage demands of grid-based probability densities, but standard algebraic approximations do not generally preserve normalization and non-negativity or account for the distribution of probability mass. This paper presents a density-aware construction of normalized and nonnegative functional low-rank representations based on the Modified Cramér–von Mises Distance. The approximation is expressed as a weighted sum of separable factors represented by Gaussian functions, Bernstein polynomials, or squared Chebyshev expansions. Its parameters and rank weights are determined via nonlinear quasi-Newton optimization using analytic gradients. Owing to the separability of the localized cumulative distribution, the cost function can be computed from samples as well as from an existing low-rank representation using only one-dimensional contractions, without assembling the full density tensor. Experiments on low-rank approximation and deterministic sample construction show that the proposed representations do preserve non-negativity and are able to capture the relevant probability mass of non-Gaussian target densities. The results demonstrate that probabilistic validity and multiscale mass agreement can be incorporated directly into low-rank density approximation.**

***Index Terms*—low-rank tensor representation, LCD-sampling, alternating least squares**

I. Introduction

Grid-based representations of probability density functions are attractive for nonlinear Bayesian estimation because they retain information about the shape of the state distribution without imposing a restrictive parametric model. Their direct application to high-dimensional state spaces, however, is limited by the exponential growth of the number of grid points. Tensor decompositions provide a means of alleviating this curse of dimensionality by replacing the dense multidimensional array by a structured collection of low-dimensional factors [1]. This principle has been applied to grid-based filtering [2], the tensor-based solution of the Fokker–Planck equation [3, 4], and nonlinear target tracking [5, 6].

Among the available tensor formats, the CANDECOMP/PARAFAC Decomposition (CPD) is particularly appealing due to its simple sum-of-products structure. Many numerical operations can be performed in terms of its one-dimensional loading vectors, and it has a low memory footprint. Nevertheless, constructing a low-rank approximation remains a nontrivial task; in fact, many tensor problems are NP-hard [7]. Standard decomposition algorithms commonly optimize an algebraic error criterion such as the Frobenius norm. When the represented tensor is a probability density, this criterion does not account for normalization, non-negativity, or the spatial distribution of probability mass. In particular, signed loading vectors can produce negative entries in the reconstructed density, an issue also observed in tensor-based density approximation [8].

Black-box and cross-approximation methods [9–11] offer an alternative when the target can be evaluated point-wise. They construct tensor representations from a limited set of adaptively selected entries and can therefore avoid assembling the full tensor. Their accuracy, however, depends on the selected evaluations resolving the relevant structure of the function. Moreover, matching selected function values does not by itself enforce normalization or non-negativity, nor does it explicitly measure agreement of localized probability mass. These observations motivate an approximation procedure that retains the computational advantages of separable representations while comparing densities according to their probability mass rather than exclusively according to point-wise values.

In this work, this comparison is based on the Modified Cramér–von Mises Distance (MCvM). The underlying Localized Cumulative Distributions (LCDs) describe probability mass locally over a range of spatial scales and provide a basis for a multivariate distance between distributions [12]. Related LCD-based criteria have also been used to construct deterministic approximations of continuous distributions [13] and of the Cartesian product between state and noise samples [14]. We combine this density-aware criterion with a structured functional CPD whose one-dimensional factors are represented by Gaussian functions, Bernstein polynomials, or squared Chebyshev expansions. The factors and rank weights are parameterized such that the resulting approximation is normalized and nonnegative by construction.

The main contributions are threefold. First, we formulate the construction of a functional low-rank density as an MCvM minimization problem for targets given either by samples or by an existing low-rank representation. Second, we derive separable objective evaluations that require only one-dimensional contractions and do not assemble the full density tensor

or its multidimensional LCD. Third, we provide positivity-preserving Gaussian, Bernstein, and Chebyshev parameterizations with analytic gradients and compare their behavior with standard and positivity-constrained Alternating Least Squares (ALS) approximations. The evaluation illustrates the resulting trade-off between point-wise approximation accuracy and agreement of localized probability mass. An overview of the proposed construction is shown in Fig. 1.

II. Problem Formulation

Let $\mathcal{X} \in \mathbb{R}^{n_1 \times \cdots \times n_N}$ denote an N-dimensional tensor. It can be approximated via a rank-R CPD given by

$$\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \circ \cdots \circ \mathbf{a}_r^{(N)}, \tag{1}$$

where $\mathbf{a}_r^{(d)} \in \mathbb{R}^{n_d}$ denotes the loading vector of rank component r in dimension d [1]. With the constraint to nonnegative elements summing up to one, this can be seen as a probability tensor and we will use it to represent arbitrary probability densities in discretized form.

The non-negativity constraint is often neglected, but negative contributions may affect subsequent probabilistic processing like performing Bayesian updates or solving the Fokker–Planck equation [4, 6, 15–17]. Correcting them after propagation introduces an additional approximation step and may increase the required rank. This motivates incorporating nonnegativity directly into the construction of the approximation.

Given a target density f, represented either by samples or in low-rank form, our objective is to construct a normalized, nonnegative approximation with a prescribed number of separable components. We determine this approximation by minimizing the MCvM [13], rather than a point-wise tensor error.

The approximation itself is described by the structured functional CPD

$$\widetilde{f}(\mathbf{x}; \boldsymbol{\theta}) = \sum_{r=1}^{M} \alpha_r \prod_{d=1}^{N} g_r^{(d)}\left(x_d; \boldsymbol{\vartheta}_r^{(d)}\right), \tag{2}$$

where M denotes the rank of the learned representation, $g_r^{(d)}$ is a parameterized one-dimensional loading function, and $\boldsymbol{\vartheta}_r^{(d)}$ denotes its family-specific parameters. The complete optimization vector $\boldsymbol{\theta}$ contains all loading-function parameters together with unconstrained logits that determine the rank weights α_r. It is specified explicitly below for each implemented function family. We consider Gaussian, Bernstein, and squared Chebyshev loading functions, whose parameterizations are defined in the next section.

The resulting optimization problem is therefore

$$\boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}} D_{\mathrm{MCvM}}\left(f, \widetilde{f}(\cdot; \boldsymbol{\theta})\right), \tag{3}$$

subject to the constraints imposed by the selected functional representation.

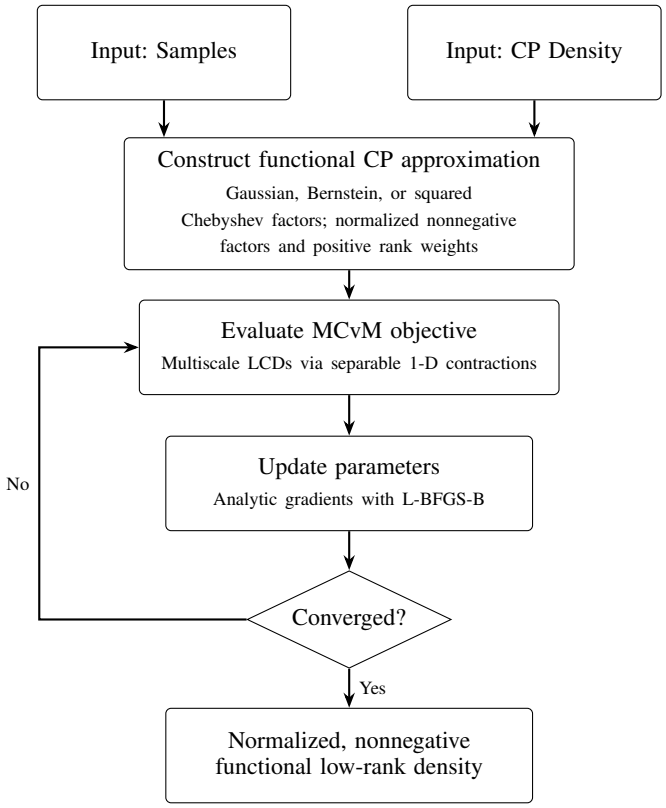


Fig. 1: Overview of the proposed MCvM-based construction of a functional low-rank density.

III. Density Representation and Objective

The proposed approach employs the MCvM as a density-aware approximation criterion. Instead of comparing the values of two densities point-wise, the distance compares localized probability masses over different spatial scales. This section details the MCvM objective-evaluation stage shown in Fig. 1.

Let $\mathbf{x} \in \mathbb{R}^N$ be a random vector with a Probability Density Function (PDF) $f(\mathbf{x}) : \mathbb{R}^N \to \mathbb{R}_+$. Its LCD is defined as

$$F(\mathbf{m}, b) = \int_{\mathbb{R}^N} f(\mathbf{x}) K_b(\mathbf{x} - \mathbf{m}) \, \mathrm{d}\mathbf{x}, \tag{4}$$

where $\mathbf{m} \in \mathbb{R}^N$ denotes the kernel center and $b \in \mathbb{R}_+$ its width [12].

In this work, a separable Gaussian kernel is employed,

$$K_b(\mathbf{x} - \mathbf{m}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{m}\|_2^2}{2b^2}\right), \tag{5}$$

which can equivalently be written as

$$K_b(\mathbf{x} - \mathbf{m}) = \prod_{d=1}^{N} \exp\left(-\frac{(x_d - m_d)^2}{2b^2}\right). \tag{6}$$

The kernel width b determines the spatial scale on which the probability mass is evaluated. Small values of b emphasize local structures, whereas larger values describe the global distribution of probability mass.

Given two densities f and $\widetilde{f}$ with corresponding LCDs F and $\widetilde{F}$, respectively, the MCvM is defined as

$$D_{\mathrm{MCvM}} = \int_{\mathbb{R}_+} w(b) \int_{\mathbb{R}^N} \left(\widetilde{F}(\mathbf{m}, b) - F(\mathbf{m}, b)\right)^2 \mathrm{d}\mathbf{m}\, \mathrm{d}b. \tag{7}$$

For the multiscale comparison, we employ

$$w(b) = b^{1-N}. \tag{8}$$

In the numerical implementation, the scale integration is restricted to $b \in [b_{\min}, b_{\max}]$. The interval is represented by geometrically spaced scales $\{b_\ell\}_{\ell=1}^{N_b}$, and the remaining one-dimensional integral is approximated by the trapezoidal rule. With trapezoidal scale weights $q_\ell^{(b)}$, the implemented objective is proportional to

$$\widehat{D}_{\mathrm{MCvM}} = \frac{1}{2} \sum_{\ell=1}^{N_b} q_\ell^{(b)} b_\ell^{1-N} D_{b_\ell}. \tag{9}$$

The factor $1/2$ is a constant scaling and therefore does not affect the minimizer.

A. Structured Functional CP Representation

We use the structured functional CPD in (2). The target rank and the approximation rank need not coincide. In the following, Q denotes the rank of an input CPD, whereas M denotes the number of rank-one components of the learned approximation. The approximation-construction step in Fig. 1 is instantiated using Gaussian, Bernstein, or squared Chebyshev loading functions.

For all implemented representations, the rank weights are parameterized by an unconstrained vector $\boldsymbol{\eta} = [\eta_1, \ldots, \eta_{M-1}]^\mathsf{T} \in \mathbb{R}^{M-1}$. For $M > 1$, define $\mathbf{z} = [\eta_1, \ldots, \eta_{M-1}, 0]^\mathsf{T}$. The mixture weights are then obtained from the fixed-last softmax

$$\alpha_r = \frac{\exp(z_r)}{\sum_{j=1}^{M} \exp(z_j)}, \qquad r = 1, \ldots, M. \tag{10}$$

Thus, $\alpha_r > 0$, $\sum_{r=1}^{M} \alpha_r = 1$. For $M = 1$, the implementation uses $\alpha_1 = 1$ and no weight logit.

The loading functions are selected such that the probability mass represented by the low-rank approximation agrees with that of the target density over multiple spatial scales.

1) Gaussian Representation: A compact parametric density representation is obtained from Gaussian components with diagonal covariance matrices,

$$\widetilde{f}_{\mathrm{G}}(\mathbf{x}; \boldsymbol{\theta}_{\mathrm{G}}) = \sum_{r=1}^{M} \alpha_r \mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r), \tag{11}$$

with

$$\boldsymbol{\Sigma}_r = \mathrm{diag}\big(\sigma_{r,1}^2, \ldots, \sigma_{r,N}^2\big). \tag{12}$$

Since a Gaussian density with diagonal covariance is separable,

$$\mathcal{N}(\mathbf{x}; \boldsymbol{\mu}_r, \boldsymbol{\Sigma}_r) = \prod_{d=1}^{N} \mathcal{N}\big(x_d; \mu_{r,d}, \sigma_{r,d}^2\big), \tag{13}$$

the Gaussian mixture directly yields a functional CPD of rank M.

The standard deviations are optimized in logarithmic coordinates. Define $s_{r,d} = \log \sigma_{r,d}$, and collect the means and log standard deviations in

$$\mathbf{M}_\mu = [\mu_{r,d}] \in \mathbb{R}^{M \times N}, \qquad \mathbf{S} = [s_{r,d}] \in \mathbb{R}^{M \times N}. \tag{14}$$

The optimization vector used by the Gaussian solver is

$$\boldsymbol{\theta}_{\mathrm{G}} = \begin{bmatrix} \mathrm{vec}(\mathbf{M}_\mu) \\ \mathrm{vec}(\mathbf{S}) \\ \boldsymbol{\eta} \end{bmatrix} \in \mathbb{R}^{2MN+M-1}. \tag{15}$$

For $M = 1$, the last block is empty. The physical standard deviations and mixture weights are recovered according to $\sigma_{r,d} = \exp(s_{r,d})$ and (10), respectively.

The means are optimized directly with box constraints given by the represented spatial domain, $x_{d,\min} \le \mu_{r,d} \le x_{d,\max}$, while the log standard deviations satisfy $\log \sigma_{\min} \le s_{r,d} \le \log \sigma_{\max}$. Hence, $\sigma_{r,d} > 0$ is guaranteed by the parameterization, and the softmax automatically enforces positive normalized rank weights.

For a target given as a discrete CPD, the one-dimensional Gaussian factors are evaluated on the corresponding grids and normalized numerically before the CP contractions are formed. For a target given by samples, the continuous univariate LCDs of the target and of the Gaussian factors are formed, and the inner products (Dirac–Gaussian and Gaussian–Gaussian LCDs) are evaluated analytically in the continuous domain; the returned CP factors are subsequently normalized on the requested output grids.

The resulting approximation is a Gaussian mixture whose component means, diagonal covariance matrices, and weights are determined by deterministic optimization rather than random sampling.

2) Bernstein Polynomial Representation: A flexible positive representation is obtained from normalized Bernstein basis functions. For the interval $[x_{d,\min}, x_{d,\max}]$, define

$$t_d = \frac{x_d - x_{d,\min}}{x_{d,\max} - x_{d,\min}} \in [0, 1] \tag{16}$$

and the degree-K Bernstein polynomial

$$B_{k,K}(t) = \binom{K}{k} t^k (1-t)^{K-k}. \tag{17}$$

The implementation uses the normalized Bernstein/Beta basis densities

$$\beta_{k,K}^{(d)}(x_d) = \frac{K+1}{x_{d,\max} - x_{d,\min}} B_{k,K}(t_d), \qquad k = 0, \ldots, K. \tag{18}$$

Each basis function is nonnegative and integrates to one over the represented interval.

For every component and dimension, let $\boldsymbol{\zeta}_r^{(d)} \in \mathbb{R}^{K+1}$ denote basis logits. The normalized coefficients are

$$\rho_{r,k}^{(d)} = \frac{\exp(\zeta_{r,k}^{(d)})}{\sum_{j=0}^{K} \exp(\zeta_{r,j}^{(d)})}, \tag{19}$$

and the loading function is

$$g_r^{(d)}(x_d) = \sum_{k=0}^{K} \rho_{r,k}^{(d)} \, \beta_{k,K}^{(d)}(x_d). \tag{20}$$

Consequently,

$$g_r^{(d)}(x_d) \geq 0, \qquad \int_{x_{d,\min}}^{x_{d,\max}} g_r^{(d)}(x_d) \, \mathrm{d}x_d = 1. \tag{21}$$

Together with the rank-weight softmax, this yields a positive normalized low-rank density by construction.

If the Bernstein logits are collected in $\boldsymbol{\zeta} = \mathrm{vec}([\zeta_{r,k}^{(d)}])$, the corresponding optimization vector is

$$\boldsymbol{\theta}_{\mathrm{B}} = \begin{bmatrix} \boldsymbol{\zeta} \\ \boldsymbol{\eta} \end{bmatrix}. \tag{22}$$

In the implementation, the factor logits are optimized with finite box bounds for numerical robustness, whereas the rank-weight logits remain unconstrained.

3) Chebyshev Polynomial Representation: Alternatively, smooth positive loading functions are constructed from Chebyshev expansions. After mapping the spatial domain to $\xi_d \in [-1, 1]$, define

$$p_r^{(d)}(\xi_d) = \sum_{k=0}^{K} c_{r,k}^{(d)} \, T_k(\xi_d), \tag{23}$$

where

$$T_k(\xi) = \cos\left(k \arccos(\xi)\right) \tag{24}$$

denotes the Chebyshev polynomial of the first kind.

The implementation enforces positivity by squaring the polynomial and normalizing the result,

$$g_r^{(d)}(x_d) = \frac{\left[p_r^{(d)}(\xi_d(x_d))\right]^2}{\displaystyle\int_{x_{d,\min}}^{x_{d,\max}} \left[p_r^{(d)}(\xi_d(x))\right]^2 \mathrm{d}x}. \tag{25}$$

Hence, each factor is nonnegative and normalized. Numerically, the normalization and the MCvM contractions are evaluated using Clenshaw–Curtis quadrature, and the returned factors are corrected to satisfy the discrete normalization on the requested output grid.

Let $\boldsymbol{c} = \mathrm{vec}([c_{r,k}^{(d)}])$ collect the polynomial coefficients. The Chebyshev optimization vector is

$$\boldsymbol{\theta}_{\mathrm{C}} = \begin{bmatrix} \boldsymbol{c} \\ \boldsymbol{\eta} \end{bmatrix}. \tag{26}$$

The coefficient vectors are internally normalized before evaluating the squared polynomials, and finite coefficient bounds are used by Limited-memory Broyden–Fletcher–Goldfarb–Shanno with Bounds (L–BFGS–B) for numerical robustness.

The different function families therefore provide different trade-offs. Gaussian factors yield a compact parametric representation with a direct probabilistic interpretation, Bernstein factors provide positive convex combinations of localized basis densities, and squared normalized Chebyshev expansions provide flexible positive approximations of smooth loading functions.

B. Optimization

The parameters of the structured low-rank representation are determined by solving (3). In the implementation, the objective is multiplied by positive constant normalization factors to improve the numerical scaling of the optimizer. Since these factors do not depend on $\boldsymbol{\theta}$, they do not change the minimizer. In the sample-based solvers, the target self term is additionally omitted during optimization because it is constant with respect to the fitted parameters.

The minimization is performed using L–BFGS–B with analytic gradients. The objective and gradient are evaluated directly from one-dimensional contractions, so neither the full multi-dimensional tensor nor the multi-dimensional LCD needs to be assembled. These contractions are derived in Section IV. These steps correspond to the parameter-update and convergence loop in Fig. 1.

For the Gaussian representation, the optimized variables are the means, log standard deviations, and weight logits as defined in (15). Therefore, the gradient has the structure

$$\nabla_{\boldsymbol{\theta}_{\mathrm{G}}} D = \begin{bmatrix} \mathrm{vec}\left(\frac{\partial D}{\partial \mathbf{M}_\mu}\right) \\ \mathrm{vec}\left(\frac{\partial D}{\partial \mathbf{S}}\right) \\ \frac{\partial D}{\partial \boldsymbol{\eta}} \end{bmatrix}. \tag{27}$$

The derivative with respect to $\boldsymbol{\eta}$ follows from the softmax Jacobian and the chain rule. Bounds are applied directly to the means and to the log standard deviations, whereas the weight logits are unconstrained.

For the Bernstein representation, derivatives are taken with respect to the basis logits $\boldsymbol{\zeta}$. If

$$\boldsymbol{\rho} = \mathrm{softmax}(\boldsymbol{\zeta}), \tag{28}$$

then the derivative of a factor follows from the softmax Jacobian. This parameterization preserves positivity and normalization throughout the optimization.

For the Chebyshev representation, the free variables are the polynomial coefficients. The factor is the normalized square of the corresponding Chebyshev expansion, and its analytic derivative includes both the derivative of the squared polynomial and the derivative of its normalization factor. The coefficient vectors are normalized internally before factor evaluation.

The proposed formulation therefore separates the approximation problem into three components: the MCvM provides the density-aware objective, the CPD provides the low-rank tensor structure, and the selected one-dimensional function family determines the parameterization and structural properties of the resulting density representation.

IV. Construction and Deterministic Sampling

A. Construction from Samples

If the target distribution is available through samples

$$\{\mathbf{x}_i\}_{i=1}^{L}, \tag{29}$$

the current implementation represents it by the equally weighted empirical Dirac mixture

$$f_{\mathrm{s}}(\mathbf{x}) = \frac{1}{L}\sum_{i=1}^{L}\delta(\mathbf{x}-\mathbf{x}_i)\,. \tag{30}$$

Insertion into (4) yields

$$F_{\mathrm{s}}(\mathbf{m},b) = \frac{1}{L}\sum_{i=1}^{L}K_b(\mathbf{x}_i-\mathbf{m})\,. \tag{31}$$

Thus, the LCD can be evaluated directly from the samples without first constructing a conventional density estimate.

For a sample-based target, we solve (3) with f replaced by the empirical Dirac mixture f_{s} in (30). For Gaussian components, the Dirac–Gaussian and Gaussian–Gaussian LCD inner products are evaluated analytically for every scale. Hence, no density grid, Kernel Density Estimation, or input CP representation is required for the optimization itself. The output grids are only used to return the fitted mixture as a discretized normalized CPD.

For the polynomial representations, the samples are contracted directly with the integrated LCD kernel, while the one-dimensional factors are evaluated on Clenshaw–Curtis quadrature nodes. In all sample-based solvers, the target self term is independent of $\boldsymbol{\theta}$ and is therefore omitted during the L–BFGS–B iterations. It is evaluated after optimization when the full MCvM value is reported.

The Gaussian components are not required to be centered at the original samples. Their positions, diagonal variances, and mixture weights are determined jointly by the optimization. Likewise, the Bernstein and Chebyshev factor parameters and their rank weights are learned jointly from the empirical sample set.

B. Construction from a Low-Rank Representation

The second case considered in this work assumes that the target density is already available in low-rank form. Before optimization, the discretized target is normalized numerically to unit total mass with respect to the employed quadrature rule. We write the target density as

$$f(\mathbf{x}) = \sum_{q=1}^{Q}\lambda_q\prod_{d=1}^{N}f_q^{(d)}(x_d), \tag{32}$$

where $f_q^{(d)}$ denotes the one-dimensional loading function in dimension d.

Due to the separability of the kernel in (5), the LCD preserves the low-rank structure,

$$F(\mathbf{m},b) = \sum_{q=1}^{Q}\lambda_q\prod_{d=1}^{N}F_q^{(d)}(m_d,b), \tag{33}$$

where

$$F_q^{(d)}(m_d,b) = \int_{\mathbb{R}} f_q^{(d)}(x_d)\exp\!\left(-\frac{(x_d-m_d)^2}{2b^2}\right)\mathrm{d}x_d. \tag{34}$$

The same property holds for the approximating functional CPD. Consequently, the spatial integral in the MCvM can be evaluated without constructing the complete N-dimensional tensor.

For a fixed value of b, define

$$D_b = \int_{\mathbb{R}^N}\Big(F(\mathbf{m},b)-\widetilde{F}(\mathbf{m},b)\Big)^2\,\mathrm{d}\mathbf{m}. \tag{35}$$

Expanding the squared difference gives

$$D_b = \langle F,F\rangle - 2\langle F,\widetilde{F}\rangle + \langle\widetilde{F},\widetilde{F}\rangle. \tag{36}$$

For two separable rank-one functions u and v,

$$\left\langle\prod_{d=1}^{N}u^{(d)},\prod_{d=1}^{N}v^{(d)}\right\rangle = \prod_{d=1}^{N}\left\langle u^{(d)},v^{(d)}\right\rangle. \tag{37}$$

Let $\mathbf{A}^{(d)}\in\mathbb{R}^{n_d\times Q}$ denote the discretized factors of the target representation and $\mathbf{B}^{(d)}\in\mathbb{R}^{n_d\times M}$ the factors of the structured approximation.

Integrating the product of two Gaussian LCD kernels with respect to their center yields the one-dimensional kernel

$$H_b(x,y) = \sqrt{\pi}\,b\exp\!\left(-\frac{(x-y)^2}{4b^2}\right). \tag{38}$$

Let $x_i^{(d)}$ and $q_i^{(d)}$ denote the quadrature nodes and weights in dimension d. The corresponding quadrature matrix is

$$[\mathbf{H}_b^{(d)}]_{ij} = q_i^{(d)}H_b\Big(x_i^{(d)},x_j^{(d)}\Big)\,q_j^{(d)}. \tag{39}$$

Using this matrix, the one-dimensional Gram matrices are given by

$$\mathbf{G}_{ff}^{(d)}(b) = \mathbf{A}^{(d)\mathsf{T}}\mathbf{H}_b^{(d)}\mathbf{A}^{(d)}, \tag{40}$$

$$\mathbf{G}_{f\widetilde{f}}^{(d)}(b) = \mathbf{A}^{(d)\mathsf{T}}\mathbf{H}_b^{(d)}\mathbf{B}^{(d)}, \tag{41}$$

$$\mathbf{G}_{\widetilde{f}\widetilde{f}}^{(d)}(b) = \mathbf{B}^{(d)\mathsf{T}}\mathbf{H}_b^{(d)}\mathbf{B}^{(d)}. \tag{42}$$

In the following, the symbol $\star$ denotes the Hadamard (i.e., elementwise) matrix product,

$$[\mathbf{A}\star\mathbf{B}]_{ij} = A_{ij}B_{ij}. \tag{43}$$

The multidimensional Gram matrices can therefore be obtained from the Hadamard products of the corresponding one-dimensional matrices,

$$\mathbf{G}_{ff}(b) = \bigstar_{d=1}^{N}\mathbf{G}_{ff}^{(d)}(b), \tag{44}$$

$$\mathbf{G}_{f\widetilde{f}}(b) = \bigstar_{d=1}^{N}\mathbf{G}_{f\widetilde{f}}^{(d)}(b), \tag{45}$$

$$\mathbf{G}_{\widetilde{f}\widetilde{f}}(b) = \bigstar_{d=1}^{N}\mathbf{G}_{\widetilde{f}\widetilde{f}}^{(d)}(b). \tag{46}$$

Consequently,

$$\begin{aligned}D_b = {}&\boldsymbol{\lambda}^{\mathsf{T}}\mathbf{G}_{ff}(b)\boldsymbol{\lambda}\\ &-2\boldsymbol{\lambda}^{\mathsf{T}}\mathbf{G}_{f\widetilde{f}}(b)\boldsymbol{\alpha}\\ &+\boldsymbol{\alpha}^{\mathsf{T}}\mathbf{G}_{\widetilde{f}\widetilde{f}}(b)\boldsymbol{\alpha}.\end{aligned} \tag{47}$$

The complete distance follows from the remaining one-dimensional integral

$$D_{\mathrm{MCvM}} = \int_{b_{\min}}^{b_{\max}} w(b) D_b \, \mathrm{d}b, \tag{48}$$

which is evaluated numerically according to (9). For the Gaussian CP solver, the spatial integrals are discretized with trapezoidal quadrature on the supplied grids. For the polynomial CP solvers, the target is interpolated to Clenshaw–Curtis nodes and the contractions use the associated Clenshaw–Curtis weights.

The main advantage of this formulation is that neither the original density tensor nor its approximation has to be assembled explicitly. Instead, the evaluation is based on one-dimensional factor matrices and comparatively small Gram matrices. Hence, the density-aware distance can be evaluated while retaining the favorable scaling properties of the low-rank representation.

The multi-dimensional products required for the objective and gradient are formed as Hadamard products of one-dimensional Gram matrices. Products excluding one dimension are evaluated with prefix/suffix Hadamard products, avoiding a repeated $\mathcal{O}(N^2)$ construction over dimensions. For a low-rank target, parameter-independent target contractions are precomputed and reused during the L–BFGS–B iterations.

C. Deterministic Sampling from a Low-Rank Representation

Eberhardt et al. [14] minimize the MCvM to construct Dirac approximations of products of independent Dirac mixtures, exploiting separability without explicitly assembling the joint density. Here, we apply the same principle to a target in CP format, whose sum-of-products structure also permits dependence between coordinates. Let the normalized target be given by (32). We approximate it by the equally weighted Dirac mixture

$$\widetilde{f}_{\mathrm{D}}(\mathbf{x}; \mathbf{Y}) = \sum_{i=1}^{L} \omega_i \delta(\mathbf{x} - \mathbf{y}_i), \qquad \omega_i = \frac{1}{L}, \tag{49}$$

where $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_L]^{\mathsf{T}} \in \mathbb{R}^{L \times N}$ collects the point locations. Only the positions are optimized; the point weights remain fixed and therefore preserve unit total mass. The corresponding LCD is

$$\widetilde{F}_{\mathrm{D}}(\mathbf{m}, b) = \sum_{i=1}^{L} \omega_i K_b(\mathbf{y}_i - \mathbf{m}). \tag{50}$$

After integration over the LCD center, all spatial terms can again be expressed through the kernel H_b from (38). For target component q and point i, define the one-dimensional target–point contraction

$$[\mathbf{C}^{(d)}(b)]_{qi} = \sum_{j=1}^{n_d} q_j^{(d)} A_{jq}^{(d)} H_b\Big(x_j^{(d)}, y_{i,d}\Big), \tag{51}$$

and combine the dimensions by

$$\mathbf{C}(b) = \bigstar_{d=1}^{N} \mathbf{C}^{(d)}(b) \in \mathbb{R}^{Q \times L}. \tag{52}$$

Thus, $C_{qi}(b)$ is the integrated LCD inner product between target rank-one component q and Dirac point i. The point–point matrix is available in closed form,

$$\begin{aligned}[\mathbf{P}(b)]_{ij} &= \prod_{d=1}^{N} H_b(y_{i,d}, y_{j,d}) \\ &= (\sqrt{\pi} b)^N \exp\left(-\frac{\|\mathbf{y}_i - \mathbf{y}_j\|_2^2}{4b^2}\right).\end{aligned} \tag{53}$$

Let $\boldsymbol{\omega} = [\omega_1, \ldots, \omega_L]^{\mathsf{T}}$ and denote the target self term by

$$T_b = \boldsymbol{\lambda}^{\mathsf{T}} \mathbf{G}_{ff}(b) \boldsymbol{\lambda}. \tag{54}$$

Expanding the squared LCD difference as in (36) then yields

$$D_b^{\mathrm{D}}(\mathbf{Y}) = T_b - 2\boldsymbol{\lambda}^{\mathsf{T}} \mathbf{C}(b) \boldsymbol{\omega} + \boldsymbol{\omega}^{\mathsf{T}} \mathbf{P}(b) \boldsymbol{\omega}. \tag{55}$$

The first term is independent of the point locations and is precomputed for all scales. The second term attracts the point set toward the probability mass of the target, whereas the third term accounts for the interaction between the Dirac points and prevents all points from collapsing onto the same location.

For numerical scaling, the implementation normalizes the multiscale distance by the weighted target self term

$$D_{\mathrm{ref}} = \frac{1}{2} \sum_{\ell=1}^{N_b} q_\ell^{(b)} b_\ell^{1-N} T_{b_\ell}. \tag{56}$$

The objective minimized over the point locations is therefore

$$J_{\mathrm{D}}(\mathbf{Y}) = \frac{1}{2 D_{\mathrm{ref}}} \sum_{\ell=1}^{N_b} q_\ell^{(b)} b_\ell^{1-N} D_{b_\ell}^{\mathrm{D}}(\mathbf{Y}). \tag{57}$$

This positive normalization does not change the minimizing point set and makes objective values less dependent on the scale of the discretized target.

The gradient is also obtained from one-dimensional contractions. Differentiating (51) with respect to coordinate $y_{i,d}$ gives

$$[\dot{\mathbf{C}}^{(d)}(b)]_{qi} = \sum_{j=1}^{n_d} q_j^{(d)} A_{jq}^{(d)} H_b\Big(x_j^{(d)}, y_{i,d}\Big) \frac{x_j^{(d)} - y_{i,d}}{2b^2}. \tag{58}$$

Using products that exclude dimension d, define

$$E_{qi}^{(d)}(b) = [\dot{\mathbf{C}}^{(d)}(b)]_{qi} \prod_{k \neq d} [\mathbf{C}^{(k)}(b)]_{qi}. \tag{59}$$

Furthermore,

$$\frac{\partial P_{ij}(b)}{\partial y_{i,d}} = -P_{ij}(b) \frac{y_{i,d} - y_{j,d}}{2b^2}. \tag{60}$$

Accounting for the symmetry of the point–point term, the objective derivative is

$$\begin{aligned}\frac{\partial J_{\mathrm{D}}}{\partial y_{i,d}} = \frac{1}{D_{\mathrm{ref}}} \sum_{\ell=1}^{N_b} q_\ell^{(b)} b_\ell^{1-N} \omega_i \Bigg(&- \sum_{q=1}^{Q} \lambda_q E_{qi}^{(d)}(b_\ell) \\ &- \sum_{j=1}^{L} \omega_j P_{ij}(b_\ell) \frac{y_{i,d} - y_{j,d}}{2b_\ell^2} \Bigg).\end{aligned} \tag{61}$$

The optimization vector is $\boldsymbol{\theta}_{\mathrm{D}} = \mathrm{vec}(\mathbf{Y})$. Each coordinate is constrained to the represented domain,

$$x_{d,\min} \le y_{i,d} \le x_{d,\max}, \qquad i = 1, \dots, L, \tag{62}$$

and the resulting box-constrained problem is solved with L–BFGS–B. A well-spread Halton design, mapped to the Cartesian product of the grid intervals, provides the initial point locations. The target self terms are precomputed, and prefix/suffix Hadamard products are used for the dimension-excluding factors in (59). Hence, the full N-dimensional target density and its LCD are never assembled.

V. Evaluation

A. Negative Values

Fig. 2 compares the rank-3 approximations of a normalized, positive rank-6 target density constructed from a fixed mixture of six diagonal Gaussian components. The target and approximations are evaluated on an equally spaced 121×121 grid over $[-5, 5]^2$, with spacing $\Delta x = 10/120$ in both dimensions. All fitted CP factors are normalized before the density metrics are evaluated.

Both the ALS methods and the MCvM-based optimization use an iteration limit of 1000. For L–BFGS–B, the stopping tolerances are set to `gtol` $= 10^{-7}$ and `ftol` $= 10^{-12}$. The MCvM objective uses eight logarithmically spaced bandwidths,

$$b_\ell = b_{\min} \left(\frac{b_{\max}}{b_{\min}} \right)^{(\ell-1)/7}, \qquad \ell = 1, \dots, 8, \tag{63}$$

where $b_{\min} = 0.75\Delta x = 0.0625$ and $b_{\max} = 10$.

For the Chebyshev representation, each loading function is constructed from a squared and normalized expansion of degree $K = 20$. Each expansion contains 21 coefficients, giving $3 \cdot 2 \cdot 21 = 126$ polynomial coefficients across all components and dimensions. Together with the two free logits for the three CP weights, this gives 128 optimization parameters. The resulting density factors have polynomial degree at most 40.

For the Bernstein/Beta representation, the degree is $K = 50$. Each one-dimensional factor is a convex mixture of 51 normalized basis densities $\beta^{(d)}_{k,50}$, $k = 0, \dots, 50$, as defined in (18). The implementation therefore optimizes $3 \cdot 2 \cdot 51 = 306$ basis logits plus two free logits for the three CP weights, giving 308 optimization parameters in total.

While standard ALS minimizes a point-wise tensor approximation error, e.g., the L_2 norm, the proposed methods optimize the density-aware MCvM criterion, which compares the LCDs of the target and the approximation over multiple spatial scales.

Standard ALS achieves the smallest relative L_2 error of 1.30×10^{-1}. However, the approximation contains negative values down to -8.30×10^{-3}, resulting in a negative mass of 1.91×10^{-2}. Moreover, approximately $20.2\,\%$ of the grid points have negative density values. Thus, despite its favorable point-wise approximation accuracy, the ALS result does not constitute a valid probability density.

All positivity-preserving approaches eliminate negative mass completely. Table I compares their normalized LCD discrepancies, which measure agreement of localized probability mass.

The Chebyshev representation achieves the smallest normalized LCD discrepancy of 2.44×10^{-5}. This is approximately $5\,\%$ lower than the value of 2.57×10^{-5} obtained by standard ALS, while simultaneously preserving non-negativity over the complete domain. Its relative L_2 error, 2.28×10^{-1}, is larger than that of ALS. This result illustrates the different objectives of the two approaches: ALS favors point-wise accuracy, whereas the MCvM-based approximation favors agreement of the localized probability mass.

The positive ALS baseline yields a normalized LCD discrepancy of 7.44×10^{-5}, approximately 2.9 times that of unconstrained ALS. Hence, merely enforcing positivity on an L_2-oriented approximation does not necessarily preserve the distribution of probability mass as measured by the LCD.

For the Gaussian representation, the normalized LCD discrepancy is 1.06×10^{-4}, while the Bernstein representation yields 2.00×10^{-4}. Both approximations remain strictly non-negative, but their LCD agreement is lower than that of the Chebyshev representation in this example. The corresponding relative L_2 errors are 2.17×10^{-1} and 6.10×10^{-1}, respectively.

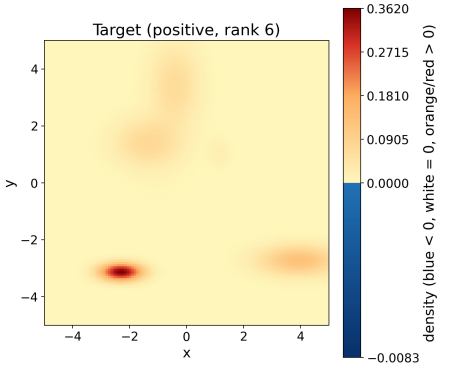
Overall, standard ALS provides the smallest relative L_2 error and the lowest computational cost, but does not yield a valid probability density. The proposed representations preserve non-negativity by construction, with their approximation quality depending on the chosen function family.

B. Construction of Samples from Low-Rank Densities

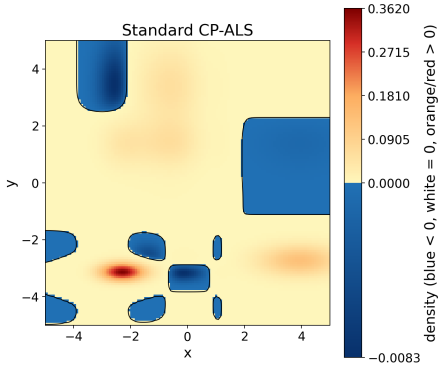
The construction of deterministic samples from a density represented in CP format is evaluated using three non-Gaussian two-dimensional target densities. The targets comprise a rank-5 mixture of separable Beta densities, a rank-7 mixture of separable Laplace densities, and an irregular rank-12 rippled spiral density. Each target is approximated by an equally weighted Dirac mixture containing $L = 60$ points. The point locations are optimized directly with respect to the normalized MCvM objective using eight spatial scales and L–BFGS–B.

Fig. 3 shows the target densities together with the optimized point locations in three side-by-side panels. From left to right, the panels show the Beta mixture, the Laplace mixture, and the rippled spiral target. In all three cases, the point sets follow the relevant mass of the target rather than merely its support. For the Beta mixture, the points cover both the broad upper mode and the more concentrated lower mode. For the Laplace mixture, they form localized groups around the individual sharp modes. The optimized points also resolve the separated and irregular structures of the rippled spiral target.

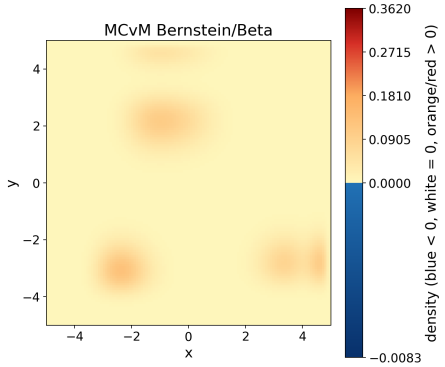
All three optimizations satisfy an L–BFGS–B convergence criterion. Table II reports the initial and final normalized MCvM objectives and the iteration counts. Relative to the initial Halton point sets, the distance is reduced by more than two orders of magnitude in every case.



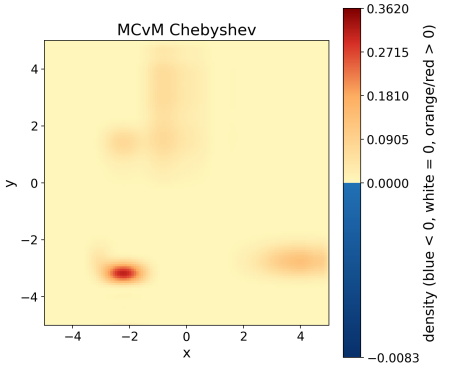
(a) 2D Gaussian mixture with six components



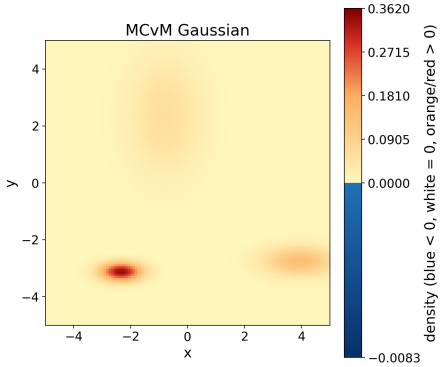
(b) Standard ALS: blue indicates negative density values



(c) MCvM-based approximation using Bernstein polynomials



(d) MCvM-based approximation using Chebyshev expansions



(e) MCvM-based approximation using Gaussian functions

Fig. 2: Target density (a), approximated using standard ALS (b) and the proposed loading-function families (c–e).

TABLE I: Rank-3 approximation metrics.

Method	Norm. LCD (10^{-5})	Time (ms)	Neg. mass (10^{-2})
ALS	2.57	0.455	1.91
Positive ALS	7.44	21.1	0
Gaussian	10.6	35.4	0
Chebyshev	2.44	444	0
Bernstein	20.0	94.9	0

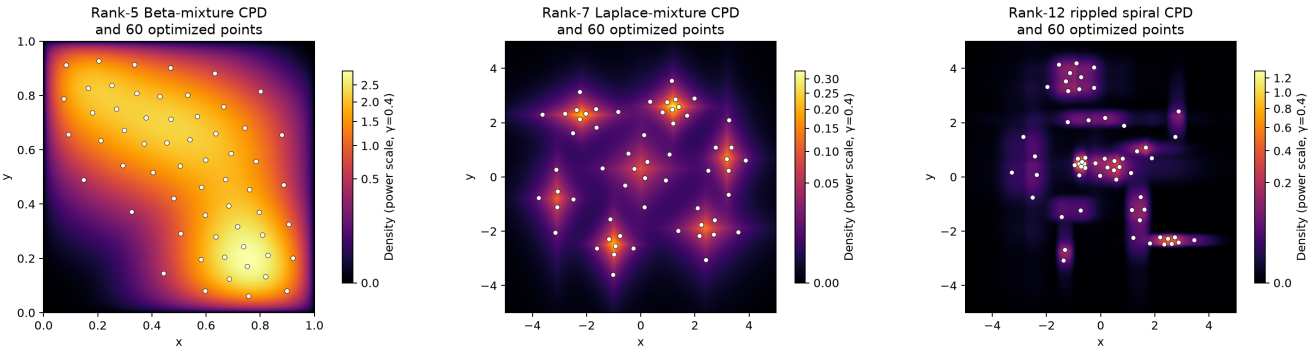


Fig. 3: Approximation of three non-Gaussian two-dimensional CP densities by equally weighted Dirac mixtures with 60 optimized points. From left to right: rank-5 Beta mixture, rank-7 Laplace mixture, and rank-12 rippled spiral density. Each panel shows the target density as a color map with the optimized point locations overlaid as white markers. The power color scale ($\gamma = 0.4$) reveals lower-density structures; colorbar ticks indicate density values.

These examples demonstrate that the MCvM formulation can construct a compact deterministic sample representation directly from a non-Gaussian low-rank density. In particular, neither random sampling from the individual CP components nor assembly of the full density tensor is required.

VI. Conclusion

This work presented low-rank density approximation based on the MCvM, comparing probability mass over multiple spatial scales rather than point-wise values. Normalized Gaussian, Bernstein, and squared Chebyshev factors with positive

TABLE II: Normalized MCvM objective before and after optimization of 60 points initialized with a Halton design.

Target	Initial	Final	Iterations
Beta mixture	4.215×10^{-2}	7.622×10^{-5}	199
Laplace mixture	4.175×10^{-2}	4.611×10^{-5}	241
Rippled spiral	7.986×10^{-2}	2.591×10^{-5}	104

rank weights ensure valid nonnegative densities. The separable LCD kernel reduces the objective to products of one-dimensional contractions, allowing targets given by samples or in CP format to be processed without assembling the full density tensor or its multidimensional LCD. Analytic gradients enable joint optimization of factor parameters and rank weights using L–BFGS–B.

The numerical results illustrate the distinction between point-wise accuracy and probabilistic validity. Standard ALS achieves the smallest relative L_2 error in the rank-reduction example, but produces negative density values. The proposed representations, on the other hand, remain nonnegative, with the Chebyshev representation attaining the smallest normalized LCD discrepancy in this test. Furthermore, the CPD-to-sample experiments show that complex non-Gaussian low-rank densities can be represented by compact sets of deterministic points, with the normalized MCvM objective reduced by more than two orders of magnitude relative to the initial Halton point sets for all considered targets.

Future work will investigate higher-dimensional problems, systematic rank-versus-accuracy trade-offs, and sample-to-CPD reconstruction under varying sample sizes. Incorporating the proposed approximation into sequential Bayesian filtering will also allow its influence on accumulated estimation errors, numerical stability, and computational cost to be assessed.

REFERENCES

[1] T. G. Kolda and B. W. Bader, "Tensor Decompositions and Applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[2] J. Matoušek, M. Brandner, J. Duník, and I. Punčochář, "Tensor train discrete grid-based filters: Breaking the curse of dimensionality," *IFAC-PapersOnLine*, vol. 58, no. 15, pp. 19–24, 2024. 20th IFAC Symposium on System Identification SYSID 2024.

[3] Y. Sun and M. Kumar, "Nonlinear Bayesian Filtering Based on Fokker-Planck Equation and Tensor Decomposition," in *2015 18th International Conference on Information Fusion (Fusion)*, pp. 1483–1488, July 2015.

[4] B. Demissie, F. Govaers, and M. A. Khan, "Nonlinear Filter Design Using Fokker-Planck Propagator in Kronecker Tensor Format," in *Proceedings of the 19th International Conference on Information Fusion*, (Heidelberg, Germany), 2016.

[5] F. Govaers, B. Demissie, A. Khan, M. Ulmke, and W. Koch, "Tensor Decomposition based Multi Target Tracking in Cluttered Environments," *Journal on Advances in Information Fusion (JAIF)*, 2019.

[6] J. Gehlen, M. Ulmke, J. Springer, F. Govaers, and W. Koch, "Tensor Decomposition based Bearing-Only Target Tracking – an Analysis based on Real Data," in *2024 27th International Conference on Information Fusion (FUSION)*, pp. 1–6, 2024.

[7] C. J. Hillar and L.-H. Lim, "Most Tensor Problems Are NP-Hard," *J. ACM*, vol. 60, pp. 45:1–45:39, Nov. 2013.

[8] J. Ajgl and O. Straka, "Aspects of Density Approximation by Tensor Trains," in *2025 28th International Conference on Information Fusion (FUSION)*, pp. 1–8, 2025.

[9] D. Savostyanov and I. Oseledets, "Fast Adaptive Interpolation of Multi-Dimensional Arrays in Tensor Train Format," in *The 2011 International Workshop on Multidimensional (nD) Systems*, pp. 1–8, 2011.

[10] D. V. Savostyanov, "Quasioptimality of Maximum-Volume Cross Interpolation of Tensors," *Linear Algebra and its Applications*, vol. 458, pp. 217–244, 2014.

[11] J. Ballani, L. Grasedyck, and M. Kluge, "Black Box Approximation of Tensors in Hierarchical Tucker Format," *Linear algebra and its applications*, vol. 438, no. 2, pp. 639–657, 2013.

[12] U. D. Hanebeck and V. Klumpp, "Localized Cumulative Distributions and a Multivariate Generalization of the Cramér-von Mises Distance," in *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, pp. 33–39, IEEE, 2008.

[13] I. Gilitschenski and U. D. Hanebeck, "Efficient Deterministic Dirac Mixture Approximation of Gaussian Distributions," in *2013 American Control Conference*, pp. 2422–2427, IEEE, 2013.

[14] H. Eberhardt, V. Klumpp, and U. D. Hanebeck, "Optimal dirac approximation by exploiting independencies," in *Proceedings of the 2010 American Control Conference*, pp. 1392–1398, IEEE, 2010.

[15] J. Gehlen, F. Govaers, J. Dunik, and W. Koch, "Grid-based estimation with dynamic rank-preserving hierarchical tucker decomposition," in *Proceedings of the 29th International Conference on Information Fusion*, 2026. Paper ID 87.

[16] J. Gehlen and F. Govaers, "On a Fast CPD-Tensor Operator for Target Tracking," in *2025 28th International Conference on Information Fusion (FUSION)*, pp. 1–8, 2025.

[17] J. Matoušek, J. Duník, F. Govaers, and J. Gehlen, "Diffusion in Lagrangian Grid-Based Predictors," in *2025 28th International Conference on Information Fusion (FUSION)*, pp. 1–8, 2025.