Scalable Learning of Multivariate Distributions via Coresets

Zeyu Ding **Katja Ickstadt** **Nadja Klein** **Alexander Munteanu** **Simon Omlor**
Lamarr Institute for ML & AI, TU Dortmund
Scientific Computing Center, Karlsruhe Institute of Technology
Department of Statistics, TU Dortmund

Abstract

Efficient and scalable non-parametric or semi-parametric regression analysis and density estimation are of crucial importance to the fields of statistics and machine learning. However, available methods are limited in their ability to handle large-scale data. We address this issue by developing a novel coreset construction for multivariate conditional transformation models (MCTMs) to enhance their scalability and training efficiency. To the best of our knowledge, these are the first coresets for semi-parametric distributional models. Our approach yields substantial data reduction via importance sampling. It ensures with high probability that the log-likelihood remains within multiplicative error bounds of $(1 \pm \varepsilon)$ and thereby maintains statistical model accuracy. Compared to conventional full-parametric models, where coresets have been incorporated before, our semi-parametric approach exhibits enhanced adaptability, particularly in scenarios where complex distributions and non-linear relationships are present, but not fully understood. To address numerical problems associated with normalizing logarithmic terms, we follow a geometric approximation based on the convex hull of input data. This ensures feasible, stable, and accurate inference in scenarios involving large amounts of data. Numerical experiments demonstrate substantially improved computational efficiency when handling large and complex datasets, thus laying the foundation for a broad range of applications within the statistics and machine learning communities.

Proceedings of the 29[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

1 INTRODUCTION

In today's era of Big Data, the vast volume and increasing complexity of data in many real-world applications pose new challenges to statistics and machine learning. Traditional methods require intense computing times on large data and straightforward solutions such as uniform subsampling may lead to infeasible solutions. They often rely on strong modeling assumptions that are too restrictive to capture complicated phenomena accurately. These limitations demand computationally efficient and scalable techniques for more flexible multivariate models.

To address the need of scalable learning, the method of *coresets* received a lot of attention in recent years. It aims to sample or select a relatively small but representative subset of the original data that can be used to accurately approximate the objective function for any solution, thereby significantly reducing computing, memory and storage requirements (Phillips, 2017; Munteanu and Schwiegelshohn, 2018).

However, most of the existing coreset literature focused on clustering, linear regression, or generalized linear models that can handle only relatively simple and limited distributional assumptions. Little research has been conducted on more flexible non-parametric, semi-parametric or highly non-linear multivariate models. This gap becomes more critical as the machine learning and statistics communities increasingly rely on complex distributional regression and estimation methodologies including not only multivariate features but also multivariate outcomes.

1.1 Brief Introduction to MCTMs

In this paper, we specifically focus on multivariate conditional transformation models (MCTMs; Klein et al., 2022), which have the unique advantage of modeling both unconditional and conditional distributions, non-linear dependence structures, and flexible effects of features. The fundamental idea of MCTMs is to model marginal distributions non-parametrically via mono-

tonic transformation functions. Multiple univariate distributions are then combined into a multivariate distribution via a *copula*, that contributes all information required to model the multivariate dependence structure. Sklar's Theorem (Sklar, 1959) is well-known and states that every multivariate distribution can be composed and represented in that way. Of course, representing and calculating such a model explicitly in a lossless way for arbitrary multivariate distributions may require solving infinite-dimensional numerical problems which is arguably not practical.

MCTMs thus formulate a semi-parametric model that rely on the monotonic univariate transformations being approximated as a linear combination of Bernstein polynomial basis functions whose choice ensures monotonicity of the composed functions. The dependence structure between the univariate output components is imposed by a Gaussian copula (Song, 2000). We note that those modeling choices may be replaced by different functional basis families and different copulas, but we stress and demonstrate that it is not as restrictive as it may seem. Notice that, although Gaussian copulas are centrally symmetric, the resulting model need not be symmetric unless all marginal distributions are symmetric as well, see for instance the density contour plots in Figures 2 to 6 in Section E.1.2.

For the sake of a concise presentation, we restrict ourselves to unconditional density estimation without features. See (Klein et al., 2022) for details and an extension to the conditional case of distributional regression. The goal is to learn the density $f_Y(y)$ of a J-dimensional random vector $Y = (Y_1, \dots, Y_J)^T \in \mathbb{R}^J$ and its distribution function $F_Y(y) = P(Y \leq y) = P(h(Y) \leq h(y))$. This involves a bijective, strictly monotonically increasing transformation function $h : \mathbb{R}^J \to \mathbb{R}^J$ estimated from the data. It maps Y to another random vector $Z \in \mathbb{R}^J$ that follows some convenient reference distribution. In particular, the common marginal distribution P_Z of the components $Z_j \sim P_Z$, $j \in [J]$ is a modeling choice and is assumed to not depend on the data. It thus holds that

$$h(Y) = (h_1(Y), \dots, h_J(Y))^T \stackrel{d}{=} (Z_1, \dots, Z_J)^T = Z\,.$$

Furthermore, it is assumed that each component h_j can be expressed as a linear combination of strictly monotonically increasing marginal transformation functions $\tilde{h}_j : \mathbb{R} \to \mathbb{R}$ of the form

$$h_j(y_1, \dots, y_j) = \lambda_{j1}\tilde{h}_1(y_1) + \dots + \lambda_{jj}\tilde{h}_j(y_j)\,,$$

where λ_{jl}, $l \leq j$ are the unrestricted entries of a $J \times J$ lower triangular matrix Λ, such that $\Sigma = \Lambda^{-1}(\Lambda^{-1})^T$ is the covariance matrix of Z. Hence, Λ describes the conditional dependencies between the J components. The functions $\tilde{h}_j(y_j), j \in [J]$, are scaled marginal transformations, each of which carry an intercept term and together allow the generation of arbitrary marginal distributions. The marginal transformations are modeled semi-parametrically via some basis functions $a_j : \mathbb{R} \to \mathbb{R}^d$ and basis coefficients $\vartheta_j \in \mathbb{R}^d$. The a_j are chosen, for example, to be Bernstein polynomials which allow for a convenient way to impose monotonicity. The unknown model parameters are collected in the vector $\theta = (\vartheta^T, \lambda^T)^T$. For consistency with other coreset literature, we consider *minimizing* the negative log-likelihood, or specifically the sum of loss contributions of each data point $y_i \in \mathbb{R}^J, i \in [n]$, to the negative log-likelihood, which is equivalent to calculating the maximum likelihood estimator $\hat{\theta}_n = \operatorname{argmin}_{\theta=(\vartheta^T,\lambda^T)^T} f(\theta)$, where

$$f(\theta) = \sum_{i=1}^{n} \frac{1}{2} \sum_{j=1}^{J} \left(\sum_{j=1}^{j-1} \lambda_{jj} a_j(y_{ij})^T \vartheta_j + a_j(y_{ij})^T \vartheta_j \right)^2 - \log(a_j'(y_{ij})^T \vartheta_j)\,. \tag{1}$$

See (Klein et al., 2022) for details and a derivation of the log-likelihood and its gradients. MCTMs can be fitted to data by optimizing the linear basis coefficients ϑ of univariate transformation models and the covariance structure of the Gaussian copula, which is parametrized through the unrestricted entries of the lower triangular matrix of the modified Cholesky factor Λ. MCTMs can thus be seen as a multivariate extension of univariate conditional transformation models (CTMs; Hothorn et al., 2014) to handle multidimensional outputs and their dependence structure.

As can be seen in Equation (1), the log-likelihood function of an MCTM contains both quadratic and logarithmic parts in terms of the parameters to optimize. The quadratic part is numerically tractable and can be reformulated in a way that enables handling within the framework of ℓ_2-subspace embeddings (Woodruff, 2014), in particular via ℓ_2 leverage score subsampling (Drineas et al., 2006; Rudelson and Vershynin, 2007). The logarithmic part is known to be unstable and computationally burdensome when optimizing models to fit large-scale datasets. Considering sensitivity sampling for similarly structured loss functions has required special efforts and novel techniques in recent work on Poisson regression models (Lie and Munteanu, 2024). For the MCTMs considered here, the situation is even more aggravated since the terms that show up in the log-likelihood do not directly depend on the plain data, but instead on the polynomial basis transformations in the quadratic part and on their derivatives in the logarithmic part.

1.2 Our Goals and Contributions

We develop the first coreset approximation framework for MCTMs, aiming to achieve the following goals.

1. **Significant computational reduction while guaranteeing an accurate log-likelihood approximation.** By combining ℓ_2 leverage score sampling with a geometric convex-hull approximation, we provide a $(1 \pm \varepsilon)$-factor log-likelihood approximation for MCTMs using only a small fraction of the data.

2. **Solving the problem of unstable values of logarithmic terms.** We construct coresets separately for $a(y)$ that represent polynomial basis transformations of plain data y, and their respective derivatives $a'(y)$. This allows to use standard methods for the former part. For the second part, extending recent prior work of Lie and Munteanu (2024), our method includes points to avoid extreme directions where the logarithm tends to infinity or produces infeasible solutions.

3. **Compatibility with other popular probabilistic models:** MCTMs build a flexible semi-parametric framework that has connections to contemporary machine learning methods such as deep learning and normalizing flows (Kobyzev et al., 2021; Papamakarios et al., 2021). By scaling up MCTMs, our coresets bring significant speedup and scaling possibilities to downstream learning tasks that build upon MCTMs.

Theoretically, we develop a leverage score and sensitivity sampling analysis along with a convex hull approximation scheme to provide a small subset of data. We prove that it approximates the log-likelihood within a multiplicative $(1\pm\varepsilon)$ factor to the full dataset.

Empirically, we demonstrate through simulated and real-world data experiments that the method offers significant benefits for large-scale data: increased computational efficiency and scalability, preserving the original model fit, likelihood ratio, and estimation bias.

We summarize our main contributions as follows:

1. **The first coresets for the MCTM framework.** To the best of our knowledge, we are the first to develop a coreset construction method for MCTMs, filling the gap of data reduction techniques for highly flexible semi-parametric model fitting methods for multivariate distributions.

2. **Convex hull based stabilization of logarithmic normalizing terms.** Numerical issues of logarithmic terms in the log-likelihood are eliminated based on prior work (Lie and Munteanu, 2024) by a convex-hull approximation of the derivative $a'(y)$ of transformed data y . The properties of this geometric approximation are investigated theoretically and empirically.

3. **Illustrative large-scale data scenarios.** We illustrate through experiments with simulated and real-world data that our new method operates more efficiently than using the original large-scale data. At the same time it retains almost the same model fit as the original full-data estimation, illustrating our theoretical guarantees in practice.

1.3 Related Work

Probabilistic Transformation Models. Originally restricted to univariate outputs, Hothorn et al. (2014) proposed a boosting approach for estimating a monotonically invertible function that transforms the output variable to a convenient, data independent distribution (e.g., standard normal). This transformation function is a semi-parametric, monotonic spline approximation that can depend on input variables in a flexible manner. This framework indirectly models the full conditional distribution of the output and was later extended towards an entirely likelihood-based approach, known as *most likely transformations* (Hothorn et al., 2018). Klein et al. (2022) built on this approach using likelihood inference. They developed MCTMs that can estimate the joint conditional distribution of a multivariate response directly based on the features. The original work of Klein et al. (2022) considers datasets of up to ten thousand observations and up to ten-dimensional outputs. In contrast to previous probabilistic methods, MCTMs directly estimate the joint distribution function of a multivariate output vector, avoiding the loss of information that would result from modeling only the mean or quantiles.

Transformation models are closely related to normalizing flows (NFs; Kobyzev et al., 2021; Papamakarios et al., 2021), which are popular in machine learning. We elaborate on this connection in Section D. The main idea of both approaches is to apply a data-driven transformation such that the transformed variable follows a standard normal or some other convenient distribution. However, NFs transform their input via some kind of deep neural network as a black box, whereas our model implements a semi-parametric transformation using a Bernstein polynomial basis, which remains analytically tractable. NFs are most commonly employed as flexible variational distributions in machine learning, whereas our focus is on modeling the distribution of multivariate data along with their dependence structure. MCTMs have also been extended towards regression, where density estimation is conditioned on feature variables. The linear underlying structure of transformation functions of MCTMs have the great advantage of enhancing interpretability compared to the neural network approach

of NFs. For instance, in MCTMs we can directly interpret the dependencies between marginal components of the outcome and clearly separate these effects from those modeling the marginal distributions. Further, MCTMs are likelihood-based and therefore yield access to confidence intervals via bootstrapping.

Coresets for Large-Scale Statistical Modeling. The main idea of the coreset method is to select the *most important* subset of data based on their contribution to the objective function (e.g., log-likelihood or loss) and reweight them, so that training the model on this subset closely approximates the results of the original dataset. Early works on coresets focus on parametric models, such as linear regression (Clarkson, 2005; Drineas et al., 2006; Dasgupta et al., 2009) or generalized linear models (Munteanu et al., 2018; Molina et al., 2018; Munteanu et al., 2022; Lie and Munteanu, 2024; Frick et al., 2024). These also make up the bulk of work on coresets for statistical models.

Inferring whole probability distributions received relatively little attention in the coreset literature. Among them are coresets for univariate kernel density estimation (Phillips and Tai, 2018, 2020; Charikar et al., 2020). Bayesian coreset based models focus on multivariate parameter distributions rather than multivariate outcomes (Geppert et al., 2017; Huggins et al., 2016; Campbell and Broderick, 2018; Ding et al., 2024). To our knowledge, the only attempt to learn multivariate data distributions using coresets for scalability has been made in the context of dependency networks that infer inter-variable dependencies with graph structure (Molina et al., 2018). This approach is again composed of multiple parametric models.

In summary, the coreset method has shown great potential in improving the efficiency of regression models and Bayesian inference. However, we stress that these methods have to date been mainly developed for parametric models, e.g., (generalized) linear models. Very limited work has considered more complex and flexible non- or semi-parametric distribution models for multivariate outputs. Our attempt to combine coresets with MCTMs is, to the best of our knowledge, the first of its kind, leveraging the flexibility of multivariate distributional regression with efficient data reduction. Our work fills the gap in the methodology for inference of large-scale multivariate distributional models.

2 MCTM CORESET CONSTRUCTION

Considering the MCTM model whose negative log-likelihood function can be defined as in Equation (1), the goal of the coreset approach is to obtain a small subset $C \subset D$ of data, such that the approximation of the original likelihood function is bounded within a factor $(1 \pm \varepsilon)$. The full details and proofs of our construction are deferred to Section A.

After applying the basis functions a and their derivatives a' to the raw data, we can assume that for $(i,j) \in [n] \times [J]$ we are given data points $a_{ij} \coloneqq a_j(y_{ij}) \in \mathbb{R}^d$ and similarly for the derivatives we have $a'_{ij} \coloneqq a'_j(y_{ij}) \in \mathbb{R}^d$. We use $A, A' \in \mathbb{R}^{nJ \times d}$ to denote the corresponding data matrices. Additionally, we assume that there is an intercept, i.e., that for each (i,j) the first coordinate of both a_{ij} and a'_{ij} is 1 to make it consistent with the presence of intercepts in the transformation functions h defined before, in Section 1.1.

Now for $i \in [n], j \in [J]$ consider the function $f(a_{ij}, \vartheta, \lambda) = \frac{1}{2}(\sum_{k=1}^{j} \lambda_{j,k}\vartheta_k a_{ij})^2 - \log(\vartheta_j a'_{ij})$ which is the negative logarithm of the corresponding likelihood component $g(i,j) = \vartheta_j a'_{ij} \exp(-\frac{1}{2}(\sum_{k=1}^{j} \lambda_{j,k}\vartheta_k a_{ij})^2)$. We assume that for all $i \in [n], j \in [J]$ and all choices of parameters it holds that $g(i,j) \leq c$ for some constant $c \in \mathbb{R}_{\geq 1}$. This is a natural Lipschitz-type restriction ensuring that the distribution function has a smooth transition from 0 to 1 preventing sudden jumps. Note that this is equivalent to $-\ln(g(i,j)) \geq -\ln(c)$.

We further add to the nJ loss contributions a total shift of $\log \mathcal{N} = nJ(\ln(c) + 1)$. This corresponds to a normalization term $\mathcal{N}$ that ensures non-negativity and thus allows a relative approximation to be meaningful. Crucially, it does not affect the optimization because it is independent of the parameters that we optimize.

In the following, we sample with probabilities that are larger than the ℓ_2 leverage scores plus a uniform term and add the convex hull of $\{a'_{ij} \mid i \in [n], j \in [J]\}$. This yields a coreset for the loss function $f(A, \vartheta, \lambda)$ to be minimized in Equation (1). Let $w \in \mathbb{R}^{n \times J}$ be a weight vector. We split the weighted version of f into three parts (weights w_{ij} are omitted in the unweighted case):

1) squared part: $f_1(A, \vartheta, \lambda, w)$
$$= \frac{1}{2} \sum\nolimits_{(i,j) \in [n] \times [J]} w_{ij} \left(\sum_{k=1}^{j} \lambda_{j,k} \langle \vartheta_k, a_{ij} \rangle \right)^2$$
2) positive log part: $f_2(A, \vartheta, \lambda, w)$
$$= \sum\nolimits_{(i,j) \in [n] \times [J]} w_{ij} \max\{\log(\langle \vartheta_j, a'_{ij} \rangle), 0\}$$
3) negative log part: $f_3(A, \vartheta, \lambda, w)$
$$= \sum\nolimits_{(i,j) \in [n] \times [J]} w_{ij} \max\{-\log(\langle \vartheta_j, a'_{ij} \rangle), 0\}$$

1) Squared Part. We let $u_i = \sup_{\|x\|_2 = 1} \frac{|M_i x|^2}{\|Mx\|_2^2}$ for the i-th row of a matrix M denote their ℓ_2 leverage score. We show that sampling proportionally to the ℓ_2 leverage scores preserves the squared part f_1. Given

rows $a_{ij} \in \mathbb{R}^d$ where $i \in [n]$ and $j \in [J]$, we are looking for an ε-coreset, which is given by a matrix comprising a subset of rows indexed by $S \subseteq [n] \times [J]$ and corresponding weights $w_{i,j}$ for every $(i,j) \in S$, such that for all $\vartheta_1, \dots, \vartheta_J \in \mathbb{R}^d$ and $\lambda \in \mathbb{R}^{J \times J}$ it holds that

$$\left| \sum_{i=1}^{n} \sum_{j=1}^{J} \left(\sum_{k=1}^{j} \lambda_{j,k} \langle \vartheta_k, a_{ij} \rangle \right)^2 - \sum_{(i,j) \in S} w_{i,j} \left(\sum_{k=1}^{j} \lambda_{j,k} \langle \vartheta_k, a_{ij} \rangle \right)^2 \right| \leq \varepsilon \left(\sum_{i=1}^{n} \sum_{j=1}^{J} \left(\sum_{k=1}^{j} \lambda_{j,k} \langle \vartheta_k, a_{ij} \rangle \right)^2 \right) .$$

To this end, we arrange data points in a matrix such that sampling rows of this matrix yields a coreset for the function defined above. We set $B \in \mathbb{R}^{nJ \times dJ^2}$ to be the matrix whose rows equal $(b_{iJ+j})_k = a_{il}$ if $k = (j-1)J + l$ for some $l \in [J]$ and $(b_{iJ+j})_k = 0$ otherwise.

Then B consists of n vertically stacked blocks B_i, for $i \in [n]$. The i-th block is defined by

$$B_i = \begin{bmatrix} b_i & 0 & 0 & 0 & \cdots & 0 \\ 0 & b_i & 0 & 0 & \cdots & 0 \\ 0 & 0 & b_i & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & b_i \end{bmatrix} \in \mathbb{R}^{J \times dJ^2},$$

where $b_i = (b_{i1}, b_{i2}, \dots, b_{iJ})$. The idea is that for any possible parametrization, the squared part can be represented by a product of the new matrix B with the vector $\theta = (\vartheta^T, \lambda^T)^T$. An ε-subspace embedding for ℓ_2 is therefore sufficient to approximate the squared part, i.e., it yields that $\forall \theta \colon \|B'\theta\|_2^2 = (1 \pm \varepsilon)\|B\theta\|_2^2$, where B' consists of few weighted rows subsampled from B according to their ℓ_2 leverage scores (Woodruff, 2014).

Lemma 2.1. *There exists a coreset S for f_1 of size $O(J^2 d/\varepsilon^2)$ which can be computed in time $O(\mathrm{nnz}(B)\log(nJ) + \mathrm{poly}(dJ))$ and with high probability by sampling and reweighting according to the ℓ_2 leverage scores of B, where $\mathrm{nnz}(B)$ denotes the number of non-zero entries of B. We then have $|f_1(A,\vartheta,\lambda) - f_1(A(S),\vartheta,\lambda,w)| \leq \varepsilon f_1(A,\vartheta,\lambda)$, where $A(S)$ denotes the restriction of A to the indices in S.*

2) Positive Logarithmic Part. We now turn our attention to the positive logarithmic part f_2. This is going to be handled using the sensitivity framework, which generalizes the previous leverage score sampling for the ℓ_2 norm to more general families of functions. We refer to Section B for details. First of all, we bound the VC dimension of the logarithmic function. This is done in a standard way. Using strict monotonicity, the logarithmic function of the inner product can be inverted (respecting their domain and range), relating it to linear classifiers in d dimensions. The latter have a VC dimension of $d+1$, which is known from classic learning theory (Kearns and Vazirani, 1994).

The second part is bounding the total sensitivity, which is the sum of all sensitivity scores $s_i = \sup_{\vartheta,\lambda} \frac{f_2(a_{ij},\theta,\lambda)}{\sum f_2(a_{ij},\theta,\lambda)}$ of single data point contributions. To bound this value, we leverage that for all parameters λ and ϑ it holds that $\ln(\vartheta_j a'_{ij}) - \frac{1}{2}(w_{i,j}\sum_{k=1}^{j}\lambda_{j,k}\vartheta_k a_{ij})^2 \leq \ln(c)$, which allows us to relate the contribution of the logarithmic part to the squared part up to a constant $\gamma > 1$ such that $s_i \leq \gamma(u_i + 1/n)$. This allows to reuse the ℓ_2 leverage scores for this part as well, albeit with an additional uniform component, and with an increase of the sample size, which comes from incorporating the VC dimension and the Lipschitz bound c. We also note that the ε-error is relative to f_1 instead of f_2 directly.

Lemma 2.2. *Assume that S is a sample consisting of $O(J^2 d^2 \ln(cdJ) c^6/\varepsilon^2)$ points drawn with probability $p_i \geq \alpha(u_i + 1/n)$, where $\alpha \in O(J^2 d \ln(cdJ) c^6/\varepsilon^2)$. Then with high probability it holds that $|f_2(A,\vartheta,\lambda) - f_2(A(S),\vartheta,\lambda,w)| \leq \varepsilon f_1(A,\vartheta,\lambda)$, where $A(S)$ denotes the restriction of A to the indices in S.*

3) Negative Logarithmic Part. Next, we handle the remaining negative logarithmic part given by f_3. We note that it has an asymptote at 0, precluding finite bounds on the sensitivity. We handle this similarly to (Lie and Munteanu, 2024) by restricting the optimization space to $D(\eta) = \{(\vartheta,\lambda) \mid \forall (i,j) \in [n] \times [J] : \langle \vartheta_j, a'_{ij} \rangle > \eta\}$ comprising only solutions for which the inner product is bounded away by $\eta \geq 0$ from zero. Setting $\eta = 0$ corresponds to the original domain.[1] By avoiding high sensitivity points in this way, f_3 can be bounded in terms of uniform sensitivities together with the VC dimension bound $d+1$ and approximated by invoking the sensitivity framework again.

Lemma 2.3. *Let (ϑ^*, λ^*) be an optimal solution. Then there exist $(\vartheta,\lambda) \in D(\eta)$ with $|f(A,\vartheta,\lambda) - f(A,\vartheta^*,\lambda^*)| \leq 2J\eta f_1(A,\vartheta^*,\lambda^*) + J\eta n + J^2\eta^2 n$. Further, if S is a sample consisting of α points drawn with probability $p_i \geq \alpha/n$ where $\alpha \in O(d\ln(cdJ)/\eta^2)$ combined with all points that are on the convex hull of $\{a'_{ij} \mid i \in [n], j \in [J]\}$. Then with high probability, it holds for all $(\vartheta,\lambda) \in D(\eta)$ that $|f_3(A,\vartheta,\lambda) - f_3(A(S),\vartheta,\lambda,w)| \leq \eta f_1(A,\vartheta,\lambda) + \eta n$, where $A(S)$ denotes the restriction of A to the indices in S.*

Main Result. We get the following theorem by a union bound over Lemmas 2.1 to 2.3 and adding

[1] The final choice will be $\eta = \Theta(\varepsilon)$ and negative value correction to the positive domain was common practice in (Klein et al., 2022) before our theoretical investigations.

up their error bounds using the triangle inequality. The additive errors of Lemma 2.3 are further charged against the optimal cost using the normalization given by $\log \mathcal{N}$, see above at the beginning of Section 2.

Theorem 2.4. *Assume that S is a sample consisting of $O(J^2d^2\ln^3(cdJ)c^6/\varepsilon^2)$ points drawn with probability $p_i \geq \alpha(u_i + 1/n)$, where $\alpha \in O(J^2d\ln^3(cdJ)c^6/\varepsilon^2)$ combined with all extreme points $(i,j) \in [n] \times [J]$ that are on the convex hull of $\{a'_{ij} \mid i \in [n], j \in [J]\}$. Then with high probability it holds for any $(\vartheta, \lambda) \in D(\eta)$ that $|f(A,\vartheta,\lambda) - f(A(S),\vartheta,\lambda,w)| \leq \varepsilon f(A,\vartheta,\lambda)$ and there exists $(\vartheta,\lambda) \in D(\eta)$ such that $f(A,\vartheta,\lambda) \leq (1 + O(\varepsilon))f(A,\vartheta^*,\lambda^*)$, for an optimal solution (ϑ^*,λ^*).*

The formal proof can be found in Section A. We remark that the convex hull can comprise $\Omega(nJ)$ points, however, different η-kernel coresets of size $\Theta(1/\eta^{(d-1)/2})$ exist for the problem (Agarwal et al., 2004; Chan, 2004), surveyed in (Agarwal et al., 2005), in the field of computational geometry. These η-kernels also match the requirements of the shifted domain $D(\eta)$. We discuss one particular choice (Blum et al., 2019) in the experimental section 3 below.

Lower Bounds. We also give two lower bounds under different natural assumptions on the λ coefficients that define the dependence structure of the Gaussian copula. These indicate the limitations of coresets for MCTMs and show that our upper bounds leave only small gaps. The details are again in Section A.

Lemma 2.5. *There exists a dataset $\{a_{ij}\}_{i\in[n]j\in[J]}$ such that any coreset for MCTMs with $\varepsilon < 1$ has size at least $\Omega(dJ^2)$ even if $\lambda_{ij} = 0$ for all $i,j \in [n]$ with $j > i$ and $|\lambda_{ij}| \leq 1$ for all $i,j \in [J]$.*

Lemma 2.6. *There exists a dataset $\{a_{ij}\}_{i\in[n]j\in[J]}$ such that any coreset for MCTMs with $\varepsilon < 1$ has size $\Omega(dJ)$ even if $\lambda_{ii} = 1$ for $i \in [J]$ and $\lambda_{ij} = 0$ for $i < j$.*

3 EMPIRICAL EVALUATION

In this section, we systematically investigate coreset data reduction techniques for MCTMs with large-scale data for a variety of 2-dimensional data generation processes as well as two multi-dimensional, large-scale, real-world applications. Our objectives are to

(i) validate the effectiveness of the proposed sampling and convex hull algorithm for MCTMs at fitting different distributions, different correlation structures, and non-linear or heavy-tailed scenarios, and to

(ii) quantify and compare the performance of uniform sampling, pure ℓ_2 leverage score sampling, and our sensitivity sampling combined with the convex hull approximation on various metrics.

Our algorithms are stated in Section C. In particular, our sensitivity sampling is detailed in Algorithm 1. As convex hull approximation, we implemented (Blum et al., 2019) given in Algorithm 2. For *mild* data, i.e., data that admits an η-kernel below the $\Omega(1/\eta^{(d-1)/2})$ lower bound, it yields an η-kernel of size $O(k^*/\eta^2)$, where k^* is the smallest possible size.[2] All experiments were carried out on a 2021 MacBook Pro equipped with an Apple M1 Pro chip and 16 GB of RAM. The code to reproduce our experiments is available at `https://github.com/zeyudsai/mctmcoreset/`.

3.1 Simulation Study on 2-Dimensional Data

We conduct a set of 2-dimensional data simulation experiments encompassing different dependence structures to validate the advantages of our proposed coreset approach (ℓ_2-hull) over simple leverage scores (ℓ_2-only) and uniform sampling as baselines.

All experimental results can be found in Section E. In particular, complete descriptions of 14 data generation processes (DGPs) are specified in Section E.1.1. Additional density contour plots that visualize the DGPs are in Section E.1.2.

Table 1 summarizes the simulation results for five representative scenarios. The experimental results clearly show that even in the extreme case of fitting the model using only a few dozen points, our proposed coreset method achieves a satisfactory approximation. To ensure the reliability of our experimental results, we performed 10 independent repetitions of each simulation, to obtain the means and standard deviations of results.

Based on the full experimental results in Tables 3 and 4 in Section E, we conclude that out of all 14 DGPs, our proposed coreset method (ℓ_2-hull) significantly outperforms uniform subsampling in 12 scenarios, and fails to show an advantage only in two scenarios, but never falls behind. Similarly, it also outperforms the plain ℓ_2 leverage scores (ℓ_2-only). This suggests that our coreset method has a robust performance across a wide range of data dependence structures, and is particularly suitable for dealing with non-linearities, heteroscedasticity, complex dependence structures, and multimodality. Our results thus highlight the stability and universality of our method.

Shortcomings in the two remaining scenarios, namely for `t-copula` and `skew-t`, show limitations for heavy-tailed distributions, when the coreset size is fixed. These have a dense convex hull and thus require the size of the convex hull approximation to be increased in order to compensate and reduce the error.

[2] Our theoretical results show that $\eta = \Theta(\varepsilon)$ suffices, and in the experiments we simply choose $\eta = 2\varepsilon$.

Table 1: Performance comparison of coreset methods (coreset size = 30). Results show mean ± std over 10 runs. Relative improvement: avg % improvement across metrics. Best values highlighted in **bold**.

	Method	ℓ_2 err.	λ err.	LR	Imp.(%)
DGP 1	ℓ_2-hull	2.56 ± 0.7	0.44 ± 0.1	$\mathbf{1.54 \pm 0.2}$	**12.8**
	ℓ_2-only	$\mathbf{2.54 \pm 0.6}$	0.51 ± 0.1	1.65 ± 0.3	1.6
	uniform	4.91 ± 4.7	$\mathbf{0.29 \pm 0.2}$	1.94 ± 1.1	baseline
DGP 2	ℓ_2-hull	$\mathbf{1.76 \pm 0.8}$	$\mathbf{0.09 \pm 0.1}$	$\mathbf{1.03 \pm 0.0}$	**49.8**
	ℓ_2-only	2.18 ± 0.8	0.10 ± 0.1	1.05 ± 0.0	38.8
	uniform	3.08 ± 0.9	0.11 ± 0.1	1.21 ± 0.4	baseline
DGP 3	ℓ_2-hull	$\mathbf{2.60 \pm 1.0}$	$\mathbf{0.14 \pm 0.1}$	$\mathbf{1.07 \pm 0.0}$	**49.6**
	ℓ_2-only	2.76 ± 0.9	0.16 ± 0.1	1.08 ± 0.0	43.7
	uniform	4.14 ± 1.8	0.2 ± 0.1	1.42 ± 0.3	baseline
DGP 4	ℓ_2-hull	$\mathbf{2.51 \pm 2.6}$	$\mathbf{0.06 \pm 0.0}$	$\mathbf{1.33 \pm 0.5}$	**41.4**
	ℓ_2-only	4.09 ± 4.3	0.07 ± 0.0	1.58 ± 0.5	9.0
	uniform	4.11 ± 2.4	0.14 ± 0.1	1.47 ± 0.5	baseline
DGP 5	ℓ_2-hull	$\mathbf{2.07 \pm 0.6}$	$\mathbf{0.08 \pm 0.0}$	$\mathbf{1.07 \pm 0.1}$	**64.8**
	ℓ_2-only	2.65 ± 0.8	$\mathbf{0.08 \pm 0.0}$	1.09 ± 0.1	58.4
	uniform	4.47 ± 3.3	0.16 ± 0.1	1.63 ± 1.1	baseline

3.2 Real-World Data Experiments

We evaluate our method on two large-scale multivariate datasets from different real-world applications.

• **Forest Cover Data (Covertype).** The UCI Covertype dataset (Blackard, 1998) contains $n = 581\,012$ samples and 54 variables describing terrain attributes. We focus on 10 continuous variables (e.g., elevation, slope, hillshade, distances) and a subsample of size $n = 300\,000$. We report comparisons of ℓ_2-hull versus ℓ_2-only, uniform sampling on parameter ϑ error, λ error, log-likelihood ratio in Table 2. Additional baselines include ridge leverage scores (ridge-lss) and root leverage scores (root-l2). Further results and running times are presented in Figure 13 and in Section E.

• **Equity Returns.** We consider two stock-return datasets: one comprising 10 major stocks' forty-year daily returns ($n \approx 10\,000$), and another with 20 stocks. Performances at varying coreset sizes $k \in \{50, 100, 200, 300\}$ are summarized in Tables 5 and 6 in Section E, and Figure 1 visualizes the log-likelihood ratio, parameter ℓ_2 distance and λ distance.

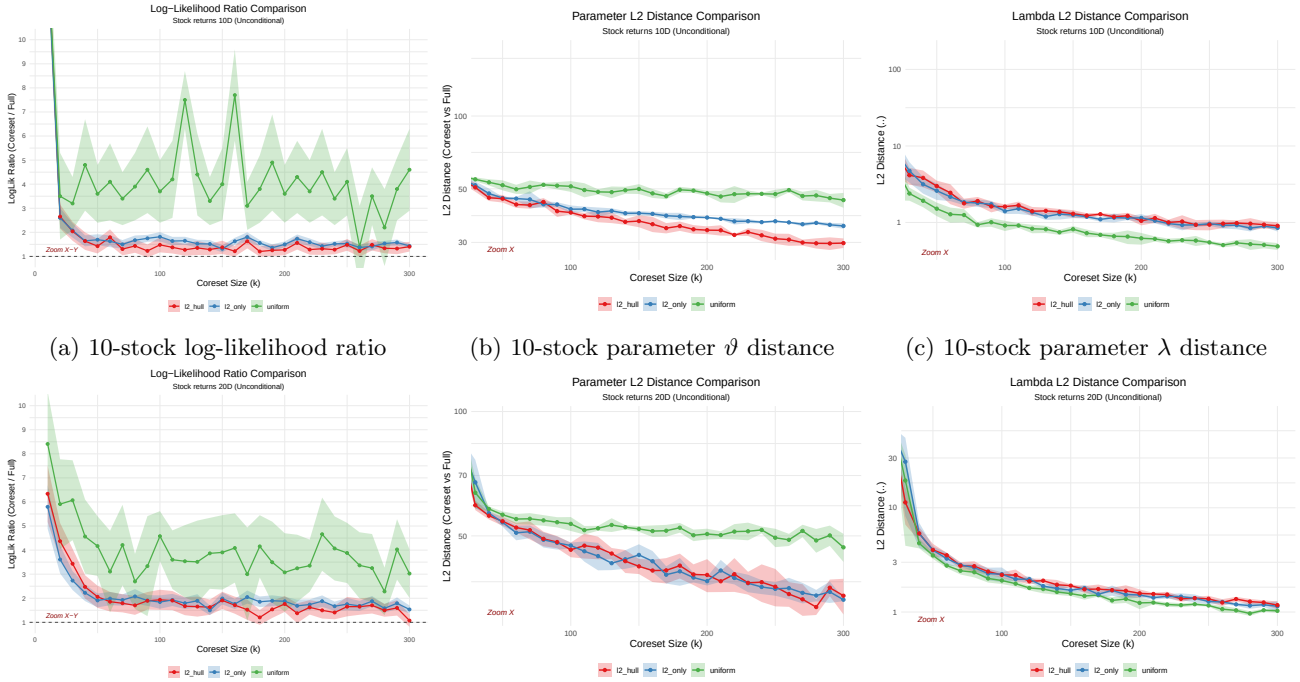
We refer to Section E.2 for further details on real-world data experiments. Across both real-world applications, our proposed ℓ_2-hull strategy outperforms uniform subsampling, and other baselines, achieving tighter log-likelihood approximations and smaller parameter errors at comparable running times. These results confirm that coresets enable scalable, accurate MCTM fitting on large, multivariate datasets.

Through experiments on the real-valued features of the Covertype dataset (as shown in Table 2), we find that the proposed coreset method (especially the hybrid sampling strategy combining ℓ_2 sensitivity with convex hull approximation) significantly outperforms uniform sampling and other baselines at different coreset sizes $k \in \{50, 100, 200, 500\}$. In particular, when the coreset size is small (e.g., $k = 50$), where the performance of uniform sampling deteriorates, our method can effectively preserve the approximation accuracy of the model, demonstrating the ability to capture important key data points. In addition, our coreset method remains stable and maintains its advantage over uniform sampling as the subset size increases. This indicates that our proposed method not only effectively reduces the computational cost on real-world datasets from several hours to a few seconds, but also accurately approximates the benchmark MCTM model fitted to the complete data, thus achieving efficient and accurate large-scale multivariate probabilistic modeling.

Table 2: Covertype data performance for different coreset sizes. Results: mean ± std over 5 trials. For ℓ_2 distances, lower is better. For log-likelihood Ratio, closer to 1 is better. Best values highlighted in **bold**.

Size	Method	Param L2	Lambda L2	LR
$k = 50$	l2-hull	$\mathbf{23.4 \pm 9.8}$	$\mathbf{18.2 \pm 12.2}$	$\mathbf{11.4 \pm 6.2}$
	l2-only	40.7 ± 24.2	37.0 ± 26.6	71.8 ± 8.7
	ridge-lss	30.8 ± 21.5	25.6 ± 24.3	51.7 ± 45.7
	root-l2	64.6 ± 10.4	63.2 ± 10.7	122.0 ± 96.1
	uniform	52.6 ± 10.6	50.6 ± 10.9	84.8 ± 90.5
$k = 200$	l2-hull	$\mathbf{14.0 \pm 3.1}$	$\mathbf{7.4 \pm 5.4}$	$\mathbf{1.42 \pm 0.13}$
	l2-only	15.7 ± 2.3	10.1 ± 4.0	1.55 ± 0.33
	ridge-lss	16.3 ± 2.2	11.5 ± 3.5	3.28 ± 5.0
	root-l2	28.2 ± 12.9	24.2 ± 15.7	14.5 ± 7.7
	uniform	37.2 ± 4.6	35.2 ± 4.8	45.1 ± 16.2
$k = 500$	l2-hull	$\mathbf{11.1 \pm 2.1}$	$\mathbf{6.5 \pm 3.5}$	$\mathbf{1.07 \pm 0.02}$
	l2-only	15.9 ± 1.2	11.7 ± 1.5	1.12 ± 0.01
	ridge-lss	16.3 ± 1.2	12.2 ± 1.5	1.27 ± 0.19
	root-l2	17.4 ± 9.0	12.7 ± 11.1	1.27 ± 0.11
	uniform	25.3 ± 10.5	21.6 ± 12.8	20.0 ± 14.0
Bench.	$n = 100k$	0	0	1
	$n = 300k$	0	0	1

Our comparisons of size vs. empirical relative errors consider different methods at fixed coreset sizes. Instead, one could evaluate the reduction in model size that our method achieves compared to its competitors at a fixed error level. However, it is impractical to fix the exact error in advance due to variability of random sampling. The reduction can be seen in Figure 1 (and in similar figures in Section E) by drawing a horizontal line at any desired error level to compare the required sample sizes. The plots thus demonstrate that to achieve the same log-likelihood ratio or parameter estimate accuracy, uniform sampling typically requires a lot larger sample size than our proposed method.



(a) 10-stock log-likelihood ratio (b) 10-stock parameter ϑ distance (c) 10-stock parameter λ distance

(d) 20-stock log-likelihood ratio (e) 20-stock parameter ϑ distance (f) 20-stock λ distance

Figure 1: Coreset performance comparison on stock-return data. Top row: 10 stocks; bottom row: 20 stocks. Shaded bands indicate ± 1 standard deviation, and solid lines show the averages over multiple repetitions.

4 DISCUSSION AND EXTENSIONS

We elaborate on limitations and possible extensions that go beyond the scope of our paper.

Choice of copula and basis functions. The Gaussian copula has been chosen as an example for *introducing* small coresets to copula models by using their close connection to ℓ_2 leverage scores. We demonstrate in our 2-dimensional examples that this already allows very flexible modeling despite a specific formulation. Our coreset method will work as well for other log-concave copulas (with convex level sets). The idea is that such distributions are *similar* to a Gaussian because we can find a so-called John ellipsoid E that is enclosed in a level set and its expansion $\sqrt{d}E$ encloses the same level set. Then, we can derive leverage scores from the quadratic form that describes the ellipsoid as in (Tukan et al., 2020), instead of the leverage scores obtained from the data. Copulas with non-convex level sets could be handled similarly, depending on the level sets being in some sense *close* to their convex hull.

Our coreset construction is agnostic to the choice of basis functions. The basis function family is thus simply exchangeable. We relied on Bernstein polynomials as in prior work by Klein et al. (2022) for convenience, as they allow to easily account for the required monotonicity constraint while being flexible enough to model *arbitrary* marginal distributions. We would thus not expect any significant difference or gains when modeling with alternative choices. However, there may be computational issues to obtain a valid CDF. For instance B-splines have been suggested by Carlan et al. (2024). Their approach is Bayesian allowing to incorporate monotonicity via appropriate prior choices, which is not directly applicable to our framework, but probably interesting towards a Bayesian extension.

Extending our methods to *conditional* transformation models would be straightforward for a linear conditional structure; it only increases the dimension dependence by the number of features conditioned on. For non-linear structures, the situation is more complicated and one would have to consider different available coreset techniques, e.g., for a specific generalized linear model such as logit, probit, or Poisson.

Dimension dependence. There are several important aspects regarding the dimension dependence.

First of all, we note that density estimation (with provable guarantees) has *principled* limitations for high-dimensional data. For instance, it is known that the empirical density approaches the original density with an error of $O(1/n^{1/d})$ in terms of Wasserstein distance (Nguyen and Ho, 2022), thus requiring a sample size of $n = O(1/\varepsilon^d)$ to bring the error term in the order of ε. Existing work is thus limited to very few dimensions.

Our 2-dimensional simulated examples are mainly meant to visually demonstrate the model's flexibility beyond Gaussian structures, despite the limitation to Gaussian copulas. Our real-world experiments are conducted on 10 variables (Covertype) and 10 resp. 20 variables (Equity returns), exceeding the usual range. We also note that for 10 dimensions and 7 basis functions, the dimension of the resulting optimization problem is significantly more than one hundred. As described in Section E.2, for such dimensions and $n \approx 580\,000$ our standard laptop crashes when fitting the full data. With our coreset reduction, we did not experience any such scalability limitations.

The coreset size specified in our theorem is $O(J^2d^2\text{polylog}(Jd))$, and we give a close lower bound of $\Omega(J^2d)$ corroborating that our methods reducing the double-sum of squares to standard methods do *not* lift to dimensions higher than necessary and do *not* impose restrictions to the spectrum as in prior work (Huang et al., 2020) on comparable loss (only w.r.t. the squared part). The additional factor d in our upper bound is an artifact of the sensitivity framework used as a black-box in our analysis; recent optimal analyses (Munteanu and Omlor, 2024a) suggest that this factor can be eliminated via tighter chaining analyses.

The convex hull component, however, has exponential $\Theta(1/\eta^{(d-1)/2})$ worst-case complexity. Unfortunately, for high-dimensional data, one must rely on their 'mildness' and to heuristics, or preselect a subset of the dimensions using leverage scores or classic PCA (Teschke et al., 2024; Pearson, 1901). In our experiments, heavy-tailed data turned out to be 'hard', for which the approximation deteriorates at fixed coreset size, and it requires an exponential increase of the convex hull component to compensate for this effect.

Data streams and distributed data. These computational settings can be realized via black-box reductions using coresets. If data are merely inserted (at possibly different sites) they can be aggregated using Merge & Reduce (cf. Geppert et al., 2020; Munteanu, 2023). To handle deletions, dynamic updates, and sparsity, one usually requires oblivious sketches (e.g., Munteanu et al., 2021, 2023; Mai et al., 2023). Leverage score sampling, convex hulls and John ellipsoids can also be approximated in data streams (Woodruff and Yasuda, 2022; Munteanu and Omlor, 2024b).

5 SUMMARY AND CONCLUSION

Our paper focuses on the *coreset* approach to enhance efficiency and scalability of fitting MCTM models to large datasets. Despite MCTM's advantages in flexibly modeling complex multivariate joint distributions, its computational burden increases significantly with increasing sample size, even with moderate output dimension. This hinders its direct application to real-world *Big Data* scenarios. To this end, we developed a novel coreset construction algorithm, which integrates subsampling based on ℓ_2 leverage scores with convex hull selection. On the one hand, the ℓ_2 leverage scores ensure that the samples focus on particularly informative observations. On the other hand, the convex hull captures the extreme value patterns of the distribution. In experiments, it has been demonstrated that our ℓ_2`-hull` algorithm reduces the data to a negligible proportion of their original size. Moreover, the fitting accuracy (log-likelihood ratio, parameter ϑ, λ distances) are virtually unchanged in contrast to pure ℓ_2 leverage score sampling and the simplest uniform sampling. Our work comprehensively evaluates fourteen 2-dimensional simulation scenarios and two multivariate real-world datasets with large sample sizes. Our findings demonstrate that ℓ_2`-hull` outperforms uniform sampling in 12 out of 14 simulated scenarios. Furthermore, it achieves significantly superior statistical performance in the context of multivariate real-world data. The running time of ℓ_2`-hull` is comparable to that of uniform sampling, yet it is neither inferior nor merely comparable to pure ℓ_2 leverage score sampling.

The main contribution of this paper is the first introduction of the coreset method to the complex semi-parametric MCTM framework. Unlike previous coresets, which were limited to parametric models such as (generalized) linear regression, we propose a new sampling scheme that naturally combines ℓ_2 leverage score sampling with iterative convex hull selection. This allows us to give a rigorous theoretical proof of the error bounds, which ensure that downstream statistical performance measures are retained when applied to large-scale data. Furthermore, we discuss the intrinsic connection between MCTMs and normalizing flows (NFs), which have become popular in machine learning recently: both of them map complex distributions to simple reference distributions through invertible transformations, thus achieving highly flexible distribution modeling. This not only provides a new perspective on the connection of semi-parametric transformation models with deep generative models (Herp et al., 2025), but also opens new research directions for inserting coreset techniques into NF models or combining MCTMs with NFs more closely in the future. Future research directions of our coreset approach include extensions to semi-parametric additive or mixed models or to Bayesian settings. Online streaming updating, and distributed calculation of our coreset construction can be explored to adapt to time- and location-varying data scenarios. Finally, it is an intriguing question how far our methods can be generalized to build coresets for NFs directly.

Acknowledgements

We thank the anonymous reviewers for their valuable comments. We thank Pia Schreiber for her preliminary work on convex hull approximations. Alexander Munteanu and Simon Omlor were supported by the German Research Foundation (DFG) – grant MU 4662/2-1 (535889065) and by the TU Dortmund – Center for Data Science and Simulation (DoDaS). Zeyu Ding and Katja Ickstadt acknowledge the support of BMBF and MKW.NRW within the Lamarr-Institute for Machine Learning and Artificial Intelligence. Nadja Klein acknowledges support by the German Research Foundation (DFG) through the Emmy Noether grant KL3037/1-1.

References

Agarwal, P. K., Har-Peled, S., and Varadarajan, K. R. (2004). Approximating extent measures of points. *J. ACM*, 51(4):606–635.

Agarwal, P. K., Har-Peled, S., Varadarajan, K. R., et al. (2005). Geometric approximation via coresets. *Combinatorial and computational geometry*, 52(1):1–30.

Blackard, J. (1998). Covertype. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C50K5N.

Blum, A., Har-Peled, S., and Raichel, B. (2019). Sparse approximation via generating point sets. *ACM Transactions on Algorithms (TALG)*, 15(3):1–16.

Campbell, T. and Broderick, T. (2018). Bayesian coreset construction via greedy iterative geodesic ascent. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, pages 698–706.

Carlan, M., Kneib, T., and Klein, N. (2024). Bayesian conditional transformation models. *Journal of the American Statistical Association*, 119(546):1360–1373.

Chan, T. M. (2004). Faster core-set constructions and data stream algorithms in fixed dimensions. In *Proceedings of the twentieth annual symposium on Computational geometry*, pages 152–159.

Charikar, M., Kapralov, M., Nouri, N., and Siminelakis, P. (2020). Kernel density estimation through density constrained near neighbor search. In Irani, S., editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 172–183. IEEE.

Clarkson, K. L. (2005). Subgradient and sampling algorithms for ℓ_1 regression. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, page 257–266.

Dasgupta, A., Drineas, P., Harb, B., Kumar, R., and Mahoney, M. W. (2009). Sampling algorithms and coresets for ℓ_p regression. *SIAM Journal on Computing*, 38(5):2060–2078.

Ding, Z., Omlor, S., Ickstadt, K., and Munteanu, A. (2024). Scalable Bayesian p-generalized probit and logistic regression. *Advances in Data Analysis and Classification*, pages 1–35.

Drineas, P., Mahoney, M. W., and Muthukrishnan, S. (2006). Sampling algorithms for l_2 regression and applications. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2006, Miami, Florida, USA, January 22-26, 2006*, pages 1127–1136. ACM Press.

Feldman, D. and Langberg, M. (2011). A unified framework for approximating and clustering data. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing (STOC)*, page 569–578.

Feldman, D., Schmidt, M., and Sohler, C. (2020). Turning Big Data into tiny data: Constant-size coresets for k-means, PCA, and projective clustering. *SIAM Journal on Computing*, 49(3):601–657.

Frick, S., Krivosija, A., and Munteanu, A. (2024). Scalable learning of item response theory models. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1234–1242.

Geppert, L. N., Ickstadt, K., Munteanu, A., Quedenfeld, J., and Sohler, C. (2017). Random projections for Bayesian regression. *Statistics and Computing*, 27(1):79–101.

Geppert, L. N., Ickstadt, K., Munteanu, A., and Sohler, C. (2020). Streaming statistical models via Merge & Reduce. *Int. J. Data Sci. Anal.*, 10(4):331–347.

Herp, M., Brachem, J., Altenbuchinger, M., and Kneib, T. (2025). Graphical transformation models. *arXiv:2503.17845*.

Hothorn, T., Kneib, T., and Bühlmann, P. (2014). Conditional transformation models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 76(1):3–27.

Hothorn, T., Möst, L., and Bühlmann, P. (2018). Most likely transformations. *Scandinavian Journal of Statistics*, 45(1):110–134.

Huang, L., Sudhir, K., and Vishnoi, N. K. (2020). Coresets for regressions with panel data. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*.

Huggins, J., Campbell, T., and Broderick, T. (2016). Coresets for scalable Bayesian logistic regression. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 29, pages 4080–4088.

Kearns, M. J. and Vazirani, U. V. (1994). *An Introduction to Computational Learning Theory*. MIT Press, Cambridge, MA, USA.

Klein, N., Hothorn, T., Barbanti, L., and Kneib, T. (2022). Multivariate conditional transformation models. *Scandinavian Journal of Statistics*, 49(1):116–142.

Kobyzev, I., Prince, S. J., and Brubaker, M. A. (2021). Normalizing flows: An introduction and review of current methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(11):3964–3979.

Langberg, M. and Schulman, L. J. (2010). Universal ε-approximators for integrals. In *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, page 598–607, USA. Society for Industrial and Applied Mathematics.

Lie, H. C. and Munteanu, A. (2024). Data subsampling for Poisson regression with pth-root-link. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Mai, T., Munteanu, A., Musco, C., Rao, A., Schwiegelshohn, C., and Woodruff, D. P. (2023). Optimal sketching bounds for sparse linear regression. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 206, pages 11288–11316.

Molina, A., Munteanu, A., and Kersting, K. (2018). Core dependency networks. In McIlraith, S. A. and Weinberger, K. Q., editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 3820–3827. AAAI Press.

Munteanu, A. (2023). Coresets and sketches for regression problems on data streams and distributed data. In *Machine Learning under Resource Constraints, Volume 1 - Fundamentals*, chapter 3.2, pages 85–97. De Gruyter, Berlin, Boston.

Munteanu, A. and Omlor, S. (2024a). Optimal bounds for ℓ_p sensitivity sampling via ℓ_2 augmentation. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*.

Munteanu, A. and Omlor, S. (2024b). Turnstile ℓ_p leverage score sampling with applications. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pages 36797–36828.

Munteanu, A., Omlor, S., and Peters, C. (2022). p-Generalized probit regression and scalable maximum likelihood estimation via sketching and coresets. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 2073–2100.

Munteanu, A., Omlor, S., and Woodruff, D. P. (2021). Oblivious sketching for logistic regression. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 7861–7871.

Munteanu, A., Omlor, S., and Woodruff, D. P. (2023). Almost linear constant-factor sketching for ℓ_1 and logistic regression. In *Proceedings of the 11th International Conference on Learning Representations (ICLR)*, pages 1–35.

Munteanu, A. and Schwiegelshohn, C. (2018). Coresets-methods and history: A theoreticians design pattern for approximation and streaming algorithms. *Künstliche Intelligenz*, 32(1):37–53.

Munteanu, A., Schwiegelshohn, C., Sohler, C., and Woodruff, D. P. (2018). On coresets for logistic regression. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NeurIPS)*, page 6562–6571.

Nguyen, K. and Ho, N. (2022). Amortized projection optimization for sliced wasserstein generative models. In *Advances in Neural Information Processing Systems 35 (NeurIPS)*.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2021). Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64.

Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572.

Phillips, J. M. (2017). Coresets and sketches. In *Handbook of Discrete and Computational Geometry*, pages 1269–1288. Chapman and Hall/CRC, 3rd edition.

Phillips, J. M. and Tai, W. M. (2018). Improved coresets for kernel density estimates. In Czumaj, A., editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 2718–2727. SIAM.

Phillips, J. M. and Tai, W. M. (2020). Near-optimal coresets of kernel density estimates. *Discret. Comput. Geom.*, 63(4):867–887.

Rudelson, M. and Vershynin, R. (2007). Sampling from large matrices: An approach through geometric functional analysis. *J. ACM*, 54(4):21.

Sklar, M. (1959). Fonctions de répartition à n dimensions et leurs marges. In *Annales de l'ISUP*, volume 8, pages 229–231.

Song, P. (2000). Multivariate dispersion models generated from Gaussian copula. *Scandinavian Journal of Statistics*, 27(2):305–320.

Teschke, S., Ickstadt, K., and Munteanu, A. (2024). Detecting interactions in high-dimensional data using cross leverage scores. *Biometrical Journal*, 66(8):e70014.

Tukan, M., Maalouf, A., and Feldman, D. (2020). Coresets for near-convex functions. In *Advances in Neural Information Processing Systems 33 (NeurIPS)*.

Woodruff, D. P. (2014). Sketching as a tool for numerical linear algebra. *Found. Trends Theor. Comput. Sci.*, 10(1-2):1–157.

Woodruff, D. P. and Yasuda, T. (2022). High-dimensional geometric streaming in polynomial space. In *63rd IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 732–743. IEEE.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes]
   Justification: The mathematical setting and the MCTM model are detailed in Section 1.1. The coreset construction framework and its underlying assumptions are described in Section 2. The final algorithms are presented in Appendix C.
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes]
   Justification: The time complexity for computing the squared part of the coreset is provided in Lemma 2.1. The sample size complexity is detailed in the main theorem and its preceding lemmas in Section 2. Time and space bounds for the approximate convex hull are given in Section 3.
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes]
   An anonymized GitHub repository link was provided in a footnote in Section 3, which contains the code to reproduce our experiments. After acceptance, the link has been de-anonymized and it will also be provided in the cam-ready submission.

2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. [Yes]
   Justification: Key assumptions are explicitly stated in Section 2. Further details are provided in the proofs.
   (b) Complete proofs of all theoretical results. [Yes]
   Justification: The main paper contains the main claims and high-level intuition of their proofs. It further explicitly states that full technical proofs are deferred to the appendix. Appendix A contains the detailed proofs for all lemmas and theorems presented in the main part.
   (c) Clear explanations of any assumptions. [Yes]
   Justification: The main assumptions are explained in context in Section 2.

3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes]
   Justification: An URL to the code repository is provided in Section 3. The data generation processes are described in detail in Appendix E.
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes]
   Justification: The experimental workflow, including coreset sizes and evaluation metrics, is described in Appendix E.
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes]
   Justification: All evaluation metrics (Likelihood Ratio, Parameter Error, etc.) are defined in Appendix E. All table and figure captions specify that metrics and results are presented as mean plus/minus std over multiple runs.
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes]
   Justification: The computing hardware is specified at the end of the introduction to Section 3.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator, if your work uses existing assets. [Yes]
   Justification: The source of the UCI Covertype dataset is cited in Section 3.2. The convex hull algorithm from (Blum et al., 2019) is also cited.
   (b) The license information of the assets, if applicable. [Not Applicable]
   (c) New assets either in the supplemental material or as a URL, if applicable. [Yes]
   Justification: We provide our source code, including data generators as a new asset via an URL in Section 3.
   (d) Information about consent from data providers/curators. [Not Applicable]
   Justification: The datasets used (UCI Covertype, stock returns) are publicly available and do not require special consent.
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]
   Justification: The datasets do not contain any personally identifiable, sensitive, or offensive content.

5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. [Not Applicable]
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Not Applicable]

Scalable Learning of Multivariate Distributions via Coresets Supplementary Materials

A THEORETICAL RESULTS

A.1 Preliminaries

Considering the MCTM model whose negative log-likelihood function can be defined as in Equation (1), the goal of the coreset approach is to obtain a subset $C \subset D$ of data, such that the approximation of the original likelihood function is bounded within a factor $(1 \pm \varepsilon)$.

After applying the basis functions a and their derivatives a' to the raw data, we can assume that for $(i,j) \in [n]\times[J]$ we are given data points $a_{ij} \in \mathbb{R}^d$ and $a'_{ij} \in \mathbb{R}^d$. We use $A, A' \in \mathbb{R}^{nJ\times d}$ to denote the corresponding data matrices. Now for $i \in [n], j \in [J]$ consider $f(a_{ij}, \vartheta, \lambda) = \frac{1}{2}(\sum_{k=1}^{j} \lambda_{j,k}\vartheta_k a_{ij})^2 - \log(\vartheta_j a'_{ij})$ which is the negative logarithm of $g(i,j) = \vartheta_j a'_{ij} \exp(-\frac{1}{2}(\sum_{k=1}^{j} \lambda_{j,k}\vartheta_k a_{ij})^2)$. We assume that for all $i \in [n], j \in [J]$ and all choices of parameters it holds that $g(i,j) \leq c$ for some constant $c \in \mathbb{R}_{\geq 1}$. This is a natural Lipschitz-type restriction ensuring that the distribution function has a smooth transition from 0 to 1 preventing sudden jumps. Note, that this is equivalent to $-\ln(g(i,j)) \geq -\ln(c)$. We further add to the nJ loss contributions a total shift of $\log \mathcal{N} = nJ(\ln(c)+1)$. This corresponds to a normalization term $\mathcal{N}$ that ensures non-negativity and thus allows a relative approximation to be meaningful, but does not affect the optimization because it is independent of the parameters that we optimize. Additionally, we assume that there is an intercept, i.e., that for each (i,j) the first coordinate of both a_{ij} and a'_{ij} is 1 to make it consistent with the presence of intercepts in the transformation functions h defined above. In the following, we show that if we sample with probabilities that are larger than the ℓ_2 leverage scores plus a uniform term and add the convex hull of $\{a'_{ij} \mid i \in [n], j \in [J]\}$, then we get a coreset for $f(A, \vartheta, \lambda)$ to be minimized as in Equation (1). Let $w \in \mathbb{R}^{n\times J}$ be a weight vector. We split the weighted version of f into three parts (w_{ij} are omitted in the unweighted case):

$$\begin{aligned}
\text{squared part:} \quad & f_1(A,\vartheta,\lambda,w) = \frac{1}{2}\sum\nolimits_{(i,j)\in[n]\times[J]} w_{ij}\left(\sum_{k=1}^{j}\lambda_{j,k}\langle\vartheta_k, a_{ij}\rangle\right)^2 \\
\text{positive log part:} \quad & f_2(A,\vartheta,\lambda,w) = \sum\nolimits_{(i,j)\in[n]\times[J]} w_{ij}\max\{\log(\langle\vartheta_j, a'_{ij}\rangle), 0\} \\
\text{negative log part:} \quad & f_3(A,\vartheta,\lambda,w) = \sum\nolimits_{(i,j)\in[n]\times[J]} w_{ij}\max\{-\log(\langle\vartheta_j, a'_{ij}\rangle), 0\}.
\end{aligned}$$

A.2 Squared Part

We let $u_i = \sup_{\|x\|_2=1} \frac{|M_i x|^2}{\|Mx\|_2^2}$ for the i-th row of a matrix M denote their ℓ_2 leverage score. We show that sampling proportionally to the ℓ_2 leverage scores preserves the squared part f_1: given rows $a_{ij} \in \mathbb{R}^d$ where $i \in [n]$ and $j \in [J]$, we are looking for an ε-coreset, which is given by a matrix comprising a subset of rows indexed by $S \subseteq [n]\times[J]$ and corresponding weights $w_{i,j}$ for every $(i,j) \in S$, such that for all $\vartheta_1, \ldots, \vartheta_J \in \mathbb{R}^d$ and $\lambda \in \mathbb{R}^{J\times J}$ it holds that

$$\begin{aligned}
&\left|\sum_{i=1}^{n}\sum_{j=1}^{J}\left(\sum_{k=1}^{j}\lambda_{j,k}\langle\vartheta_k, a_{ij}\rangle\right)^2 - \sum_{(i,j)\in S} w_{i,j}\left(\sum_{k=1}^{j}\lambda_{j,k}\langle\vartheta_k, a_{ij}\rangle\right)^2\right| \\
&\qquad\qquad \leq \varepsilon\left|\sum_{i=1}^{n}\sum_{j=1}^{J}\left(\sum_{k=1}^{j}\lambda_{j,k}\langle\vartheta_k, a_{ij}\rangle\right)^2\right|
\end{aligned}$$

In the following, we arrange data points in a matrix such that sampling rows of this matrix yields a coreset for the function defined above. We set $B \in \mathbb{R}^{nJ \times dJ^2}$ to be the matrix whose rows equal $(b_{iJ+j})_k = a_{il}$ if $k = (j-1)J + l$ for some $l \in [J]$ and $(b_{iJ+j})_k = 0$ otherwise. Then B consists of n vertically stacked blocks. The i-th block, for $i \in [n]$, is defined by

$$B_i = \begin{bmatrix} b_i & 0 & 0 & 0 & \cdots & 0 \\ 0 & b_i & 0 & 0 & \cdots & 0 \\ 0 & 0 & b_i & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & b_i \end{bmatrix} \in \mathbb{R}^{J \times dJ^2},$$

where $b_i = (b_{i1}, b_{i2}, \ldots, b_{iJ})$. The idea is that for any possible parametrization, the squared part can be represented by a product of the new matrix B with the vector $\theta = (\vartheta^T, \lambda^T)^T$. An ε-subspace embedding for ℓ_2 is therefore sufficient to approximate the squared part, i.e., it yields that $\forall \theta\colon \|B'\theta\|_2^2 = (1 \pm \varepsilon)\|B\theta\|_2^2$, where B' consists of few weighted rows subsampled from B according to their ℓ_2 leverage scores (Woodruff, 2014).

Lemma 2.1. *There exists a coreset S for f_1 of size $O(J^2 d/\varepsilon^2)$ which can be computed in time $O(\mathrm{nnz}(B)\log(nJ) + \mathrm{poly}(dJ))$ and with high probability by sampling and reweighting according to the ℓ_2 leverage scores of B, where $\mathrm{nnz}(B)$ denotes the number of non-zero entries of B. We then have $|f_1(A, \vartheta, \lambda) - f_1(A(S), \vartheta, \lambda, w)| \leq \varepsilon f_1(A, \vartheta, \lambda)$, where $A(S)$ denotes the restriction of A to the indices in S.*

Proof. We apply ℓ_2 leverage score sampling to B which gives us a set $S \subseteq [n] \times [J]$ and weights $w_{ij} \in \mathbb{R}_{\geq 1}$ for all $(i,j) \in S$ such that $|S| = O(J^2 d/\varepsilon^2)$ and with high probability for all $x \in \mathbb{R}^{dJ^2}$ it holds that

$$\left| \|Bx\|_2^2 - \sum_{(i,j)\in S} w_{i,j} \|b_{i,j} x\|_2^2 \right| \leq \varepsilon \|Bx\|_2^2.$$

Assume that the statement above holds and consider any parameterization $\vartheta_1, \ldots \vartheta_J \in \mathbb{R}^d$ and $\lambda \in \mathbb{R}^{J \times J}$. We define $x \in \mathbb{R}^{dJ^2}$ by $x_{jk} = \lambda_{j,k}\vartheta_k$. Then we have that

$$\begin{aligned} &\left| \sum_{i=1}^{n} \sum_{j=1}^{J} \left(\sum_{k=1}^{j} \lambda_{j,k} \langle \vartheta_k, a_{ij} \rangle \right)^2 - \sum_{(i,j)\in S} w_{i,j} \left(\sum_{k=1}^{j} \lambda_{j,k} \langle \vartheta_k, a_{ij} \rangle \right)^2 \right| \\ &= \left| \|Bx\|_2^2 - \sum_{(i,j)\in S} w_{i,j} \|b_{i,j} x\|_2^2 \right| \leq \varepsilon \|Bx\|_2^2 = \varepsilon \left| \sum_{i=1}^{n} \sum_{j=1}^{J} \left(\sum_{k=1}^{j} \lambda_{j,j} \langle \vartheta_k, a_{ij} \rangle \right)^2 \right|. \end{aligned}$$

□

A.3 Logarithmic Parts

We now turn our attention to the positive logarithmic part f_2. This is going to be handled using the sensitivity framework, which generalizes the previous leverage score sampling for the ℓ_2 norm to more general families of functions, we refer to Section B for details. First of all, we bound the VC dimension of the logarithmic function. This is done in a standard way. Using strict monotonicity, the logarithmic function of the inner product can be inverted (respecting their domain and range), relating it to linear classifiers in d dimensions. The latter have a known VC dimension of $d+1$ using classic learning theory results (Kearns and Vazirani, 1994). The second part is bounding the total sensitivity, which is the sum of all sensitivity scores $s_i = \sup_{\vartheta,\lambda} \frac{f_2(a_{ij},\theta,\lambda)}{\sum f_2(a_{ij},\theta,\lambda)}$ of single data point contributions. To bound this value, we leverage that for all parameters λ and ϑ it holds that $\ln(\vartheta_j a'_{ij}) - \frac{1}{2}(w_{i,j} \sum_{k=1}^{j} \lambda_{j,k}\vartheta_k a_{ij})^2 \leq \ln(c)$, which allows us to relate the contribution of the logarithmic part to the squared part up to a constant $\gamma > 1$ such that $s_i \leq \gamma(u_i + 1/n)$. This allows to reuse the ℓ_2 leverage scores for this part as well, albeit with an additional uniform component, and with an increase of the sample size, which comes from incorporating the VC dimension and the Lipschitz bound c. We also note that the ε-error is relative to f_1 instead of f_2 directly.

Lemma 2.2. *Assume that S is a sample consisting of $O(J^2 d^2 \ln(cdJ)c^6/\varepsilon^2)$ points drawn with probability $p_i \geq \alpha(u_i + 1/n)$, where $\alpha \in O(J^2 d \ln(cdJ)c^6/\varepsilon^2)$. Then with high probability it holds that $|f_2(A, \vartheta, \lambda) - f_2(A(S), \vartheta, \lambda, w)| \leq \varepsilon f_1(A, \vartheta, \lambda)$, where $A(S)$ denotes the restriction of A to the indices in S.*

Proof. Our goal is to apply the results of sensitivity sampling. For $(i,j) \in [n] \times [J]$ we define $h_{i,j}(\vartheta, \lambda) = \max\{-\ln(\vartheta_j a'_{ij}), 0\}$ and $h_0(\vartheta, \lambda) = f_1(A, \vartheta, \lambda)$. We set $h(A, \vartheta, \lambda, w) = h_0(\vartheta, \lambda) + \sum_{(i,j)\in[n]\times[J]} h_{i,j}(\vartheta, \lambda)$. It holds that the VC-dimension of $\{h_{i,j} \mid (i,j) \in [n] \times [J]\} \cup \{h_0\}$ is bounded by $d+2$ since $\log(\cdot)$ is a monotonic function and the VC dimension of $\{h_x : \vartheta \mapsto x\vartheta \mid x \in \mathbb{R}^d\}$ is bounded by $d+1$ (Kearns and Vazirani, 1994) and adding the function h_0 can only increase the VC-dimension by 1.

For the sensitivity we observe the following: since for all parameters λ and ϑ it holds that

$$\ln(\vartheta_j a'_{ij}) - \frac{1}{2}\left(\sum_{k=1}^{j} \lambda_{j,k}\vartheta_k a_{ij}\right)^2 \leq \ln(c)$$

it in particular holds that

$$\ln(\vartheta_j a'_{ij}) - \frac{1}{2}\left(\vartheta_j a_{ij}\right)^2 \leq \ln(c)$$

Note that there is some $b \in \mathbb{R}$ such that $\vartheta_j a'_{ij} = b\vartheta_j a_{ij}$. Since we can scale ϑ_j arbitrarily we get that

$$\ln(bt) - \frac{1}{2}t^2 \leq \ln(c)$$

holds for any $t \in \mathbb{R}$. Note, that the term on the left hand side is maximized if $t = 1$ and thus

$$\ln(b) - \frac{1}{2}1^2 \leq \ln(c)$$

or equivalently $b \leq ec$. We further have that the derivative of

$$\frac{\ln(bt)}{\frac{1}{2}t^2}$$

is given by $(t/2 - \ln(bt)t)/(\frac{1}{2}t^2)^2$ which is 0 if $t = 0$ or $\ln(bt) = 1/2$ or equivalently $t = \exp(\frac{1}{2})/b$ which implies that

$$\frac{\ln(bt)}{\frac{1}{2}t^2} \leq \frac{\ln(\exp(\frac{1}{2}))}{(e/2)\,(1/b)^2} = b^2/e.$$

We thus have that $s_{(i,j)} = (ec^2)u_{i,j}$ is an upper bound for the sensitivity of $h_{i,j}$. Consequently the total sensitivity is bounded by $(ec^2)dJ^2$.

Lastly we have that $h(A, \vartheta, \lambda, w) \leq (ec^2 + 1)f_1(A, \vartheta, \lambda, w)$. Thus applying sensitivity sampling with error parameter ε/c^2 gives us the desired result. □

Next, we handle the remaining negative logarithmic part given by f_3. We note that the asymptote at 0 precludes finite bounds on the sensitivity. We handle this similarly to (Lie and Munteanu, 2024) by restricting the optimization space to $D(\eta) = \{(\vartheta, \lambda) \mid \forall (i,j) \in [n] \times [J] : \langle \vartheta_j, a'_{ij} \rangle > \eta\}$ comprising only solutions for which the inner product is bounded away by $\eta \geq 0$ from zero. Setting $\eta = 0$ corresponds to the original domain.[3] By avoiding high sensitivity points in this way, f_3 can be bounded in terms of uniform sensitivities together with the VC dimension bound $d+1$ and approximated by invoking the sensitivity framework again.

Lemma 2.3. *Let (ϑ^*, λ^*) be an optimal solution. Then there exist $(\vartheta, \lambda) \in D(\eta)$ with $|f(A, \vartheta, \lambda) - f(A, \vartheta^*, \lambda^*)| \leq 2J\eta f_1(A, \vartheta^*, \lambda^*) + J\eta n + J^2\eta^2 n$. Further, if S is a sample consisting of α points drawn with probability $p_i \geq \alpha/n$ where $\alpha \in O(d \ln(cdJ)/\eta^2)$ combined with all points that are on the convex hull of $\{a'_{ij} \mid i \in [n], j \in [J]\}$. Then with high probability, it holds for all $(\vartheta, \lambda) \in D(\eta)$ that $|f_3(A, \vartheta, \lambda) - f_3(A(S), \vartheta, \lambda, w)| \leq \eta f_1(A, \vartheta, \lambda) + \eta n$, where $A(S)$ denotes the restriction of A to the indices in S.*

[3] We also note that the final choice will be $\eta = \Theta(\varepsilon)$ and negative value correction into the positive domain is common practice and was implemented in (Klein et al., 2022) before our theoretical investigations.

Proof. Let $e_1 \in \mathbb{R}^d$ be the first unit vector and let ϑ', λ be a feasible solution, i.e. $\vartheta'_j a'_{ij} > 0$. Then we claim that (ϑ, λ) with $\vartheta = \vartheta' + \eta e_1$ fulfills $f(A, \vartheta, \lambda) \leq f(A, \vartheta', \lambda) + \eta f_1(A, \vartheta', \lambda) + \eta$. Applying this to (ϑ^*, λ^*) proves the first part of the lemma. First note that as $\vartheta'_j a'_{ij} > 0$ we have that $(\vartheta, \lambda) \in D(\eta)$. Further we have that

$$f_3(A, \vartheta, \lambda) - f_2(A, \vartheta, \lambda) \leq f_3(A, \vartheta', \lambda) - f_2(A, \vartheta', \lambda)$$

as $\vartheta_j a'_{ij} \geq \vartheta'_j a'_{ij}$.

Lastly note that

$$\begin{aligned}
2f_1(A, \vartheta, \lambda) &= \sum_{(i,j) \in [n] \times [J]} \left(\sum_{k=1}^{j} \lambda_{j,k} \vartheta_k a_{ij} \right)^2 \\
&= \sum_{(i,j) \in [n] \times [J]} \left(\sum_{k=1}^{j} \lambda_{j,k} (\vartheta'_k + e_1) a_{ij} \right)^2 \\
&\leq \sum_{(i,j) \in [n] \times [J]} \left(\sum_{k=1}^{j} \lambda_{j,k} \vartheta'_k a_{ij} \right)^2 + 2J\eta \max \left\{ \left(\sum_{k=1}^{j} \lambda_{j,k} \vartheta'_k a_{ij} \right), 1 \right\} + J^2 \eta^2 \\
&\leq \sum_{(i,j) \in [n] \times [J]} \left(\sum_{k=1}^{j} \lambda_{j,k} \vartheta'_k a_{ij} \right)^2 + 2J\eta \max \left\{ \left(\sum_{k=1}^{j} \lambda_{j,k} \vartheta'_k a_{ij} \right)^2, 1 \right\} + J^2 \eta^2 \\
&= (2 + 4J\eta) f_1(A, \vartheta', \lambda) + 2J\eta n + J^2 \eta^2 n.
\end{aligned}$$

Next we show by sampling techniques that with high probability it holds that $|f_3(A, \vartheta, \lambda) - f_3(A(S), \vartheta, \lambda, w)| \leq \eta f_1(A, \vartheta, \lambda) + \eta n$. Again we use sensitivity sampling. The bound of the VC dimension is again $d + 2$. For the sensitivity we have that $-\ln(\vartheta_j a'_{ij}) \leq \eta^{-1}$. Since we only need an absolute error of ηn we can apply the results of sensitivity sampling. □

A.4 Main Result

We start with a technical lemma. With a similar proof technique as in Lemma 2.2, we get the following lower bound on the loss function that allows us to charge the previous errors depending only on f_1 and additive terms against the original loss function:

Lemma A.1. *For any $(\vartheta, \lambda) \in D(\eta)$ it holds that $f(A, \vartheta, \lambda) \geq \max\{nJ, f_1(A, \vartheta, \lambda)/(2\ln(c))\}$.*

Proof. In previous proof we showed that $\vartheta_j a'_{ij} = b\vartheta_j a_{ij}$. holds for all $(i, j \in [n] \times [J])$ for some $b \leq ec$. Consider $\frac{1}{2}t^2 - \ln(bt) + \ln(c) + 1$. For any $t \leq \mathbb{R}_{>0}$ we have that

$$\frac{1}{2}t^2 - \ln(bt) + \ln(c) + 1 \geq 1,$$

for any $t \leq \sqrt{2\ln(c)}$ we have that

$$\frac{1}{2}t^2 \leq \ln(c).$$

and thus

$$\frac{1}{2}t^2 - \ln(bt) + \ln(c) + 1 \geq \max\left\{1, \frac{1}{2}t^2 \Big/ (2\ln(c))\right\}$$

For any $t \geq \sqrt{2\ln(c)}$ we have that

$$\ln(bt) \leq \ln(ect) \leq \frac{1}{3}t^2 \leq \frac{2}{3} \cdot \frac{1}{2}t^2$$

if c is large enough and thus

$$\frac{1}{2}t^2 - \ln(bt) + \ln(c) + 1 \geq \frac{1}{2}t^2/(2\ln(c))$$

Thus, we have component-wise that

$$\frac{1}{2}w_{ij}\left(\sum_{k=1}^{j}\lambda_{j,k}\langle\vartheta_k,a_{ij}\rangle\right)^2 - w_{ij}\max\{\log(\langle\vartheta_j,a'_{ij}\rangle),0\}$$
$$\geq \max\left\{1,\frac{1}{2}w_{ij}\left(\sum_{k=1}^{j}\lambda_{j,k}\langle\vartheta_k,a_{ij}\rangle\right)^2\Big/(2\ln(c))\right\}$$

and thus the lemma follows. □

We get the following theorem by a union bound over Lemmas 2.1 to 2.3 and putting their error bounds together using the triangle inequality. The additive errors of Lemma 2.3 are further charged against the optimal cost using Lemma A.1.

Theorem 2.4. *Assume that S is a sample consisting of $O(J^2d^2\ln^3(cdJ)c^6/\varepsilon^2)$ points drawn with probability $p_i \geq \alpha(u_i+1/n)$, where $\alpha \in O(J^2d\ln^3(cdJ)c^6/\varepsilon^2)$ combined with all extreme points $(i,j)\in[n]\times[J]$ that are on the convex hull of $\{a'_{ij} \mid i\in[n], j\in[J]\}$. Then with high probability it holds for any $(\vartheta,\lambda)\in D(\eta)$ that $|f(A,\vartheta,\lambda) - f(A(S),\vartheta,\lambda,w)| \leq \varepsilon f(A,\vartheta,\lambda)$ and there exists $(\vartheta,\lambda)\in D(\eta)$ such that $f(A,\vartheta,\lambda)\leq(1+O(\varepsilon))f(A,\vartheta^*,\lambda^*)$, for an optimal solution (ϑ^*,λ^*).*

Proof. By Lemma 2.1 we have that $|f_1(A,\vartheta,\lambda) - f_1(A(S),\vartheta,\lambda,w)| \leq \varepsilon f_1(A,\vartheta,\lambda)$ with high probability.

By Lemma 2.2 we have that $|f_2(A,\vartheta,\lambda) - f_2(A(S),\vartheta,\lambda,w)| \leq \varepsilon f_1(A,\vartheta,\lambda)$ with high probability.

By Lemma 2.3 we have that $|f_3(A,\vartheta,\lambda) - f_3(A(S),\vartheta,\lambda,w)| \leq \eta f_1(A,\vartheta,\lambda) + \eta n$ with high probability.

Overall, by the triangle inequality, we have that $|f(A,\vartheta,\lambda) - f(A(S),\vartheta,\lambda,w)| \leq 2\varepsilon f_1(A,\vartheta,\lambda) + \eta f_1(A,\vartheta,\lambda) + \eta n$.

By Lemma A.1 we have that $f(A,\vartheta,\lambda)\geq\max\{nJ, f_1(A,\vartheta,\lambda)/(2\ln(c))\}$. Substituting $\varepsilon/(2J\ln(c))$ for η yields the desired bound $\forall(\vartheta,\lambda)\in D(\eta): |f(A,\vartheta,\lambda) - f(A(S),\vartheta,\lambda,w)| \leq \varepsilon f(A,\vartheta,\lambda)$

By Lemma 2.3 and using Lemma A.1 again, there exists $(\vartheta,\lambda)\in D(\eta)$ such that $f(A,\vartheta,\lambda)\leq f(A,\vartheta^*,\lambda^*) + 2\eta J f_1(A,\vartheta^*,\lambda^*) + J\eta n + J^2\eta^2 n \leq (1+O(\varepsilon))f(A,\vartheta^*,\lambda^*)$, where (ϑ^*,λ^*) is an optimal solution. □

We remark that the convex hull can comprise $\Omega(nJ)$ points, however, different η-kernel coresets of size $\Theta(1/\eta^{(d-1)/2})$ exist for the problem (Agarwal et al., 2004; Chan, 2004), surveyed in (Agarwal et al., 2005), in the field of computational geometry. These η-kernels also match the requirements of the shifted domain $D(\eta)$. We discuss our particular choice (Blum et al., 2019) in the experimental section.

A.5 Lower Bounds

It suffices to prove a lower bound for the squared part. In particular the bound will hold against preserving the subspace (equivalently the rank) spanned by the data. In the following, we give two lower bounds under different natural assumptions on the λ coefficients that define the dependence structure of the Gaussian copula. The first one is a weaker lower bound that holds without assumptions on λ other than its lower triangular structure. The second lower bound is stronger and holds under additional but common assumptions on λ.

Lemma 2.5. *There exists a dataset $\{a_{ij}\}_{i\in[n]j\in[J]}$ such that any coreset for MCTMs with $\varepsilon<1$ has size at least $\Omega(dJ^2)$ even if $\lambda_{ij}=0$ for all $i,j\in[n]$ with $j>i$ and $|\lambda_{ij}|\leq 1$ for all $i,j\in[J]$.*

Proof. Consider the instance with $n=(J-1)^2d$ consisting of Jd blocks $\{a_ij\}_{tj\in[J]}$ where $t\in J\times d$. More precisely for $t=(j_0,k)\in J\times J\times d$ we have that

$$a_{tj}=\begin{cases}e_k, & \text{if } j\geq j_0\}\\ 0 & \text{else}\end{cases}$$

We set $A \in \mathbb{R}^{j^2 d \times J^2 d}$ to be the matrix with parameters that represents these rows. For $t = (3, k)$ the t-th block $A_t \in \mathbb{R}^{J \times Jd}$ of the matrix A looks as follows:

$$A_t = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & e_k & 0 & 0 & \cdots & 0 \\ 0 & 0 & e_k & e_k & 0 & \cdots & 0 \\ 0 & 0 & e_k & e_k & e_k & \vdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & 0 \\ 0 & 0 & e_k & e_k & e_k & \cdots & e_k \end{bmatrix}$$

and the matrix itself has the form

$$A = \begin{bmatrix} A_{1,1} \\ A_{1,2} \\ \vdots \\ A_{1,J} \\ A_{2,1} \\ \vdots \\ A_{d,J} \end{bmatrix}$$

Note that the matrix $A \in \mathbb{R}^{J^2 \times d}$ with rows (a_{ij}) is of rank at least $dJ(J-1)$ even if we restrict the parameter space by requiring $\lambda_{ij} = 0$ for all $i, j \in [n]$ with $j > i$ and $|\lambda_{ij}| \leq 1$ for all $i, j \in [J]$.. To see this observe that for each $k \in [d]$ and j_1 and j_2 with $j_1 \leq j_2$ there is a parametrization such that only the contribution from row j_2 from block (k, j_1) is non zero:

$\lambda_{j_2 j_1} = 1$	
$\lambda_{j_2 (j_1 - 1)} = -1$	if $j_1 < j_2$ and $j_1 \geq 1$
$\lambda_{jl} = 0$	else
$\vartheta_k = e_k$	for $k = j_0$
$\vartheta_k = 0$	else.

Thus since any coreset coreset preserves the rank of the matrix any coreset must also be of rank at least $dJ(J-1)$ and thus of size at least $\Omega(dJ^2)$. □

Lemma 2.6. *There exists a dataset $\{a_{ij}\}_{i \in [n] j \in [J]}$ such that any coreset for MCTMs with $\varepsilon < 1$ has size $\Omega(dJ)$ even if $\lambda_{ii} = 1$ for $i \in [J]$ and $\lambda_{ij} = 0$ for $i < j$.*

Proof. Consider the instance with $n = d$ and $a_{ij} = e_i$ for all $i \in [n]$ and $j \in [J]$. Now consider any instance $\{a'_{ij}\}_{i \in [n], j \in [J]}$ with at least one zero entry, i.e. $a'_{i_0 j_0} = 0$ for some $i_0 \in [n], j_0 \in [J]$. Then A' cannot be a coreset of A for the following reason: let $A' \in \mathbb{R}^{n \times d}$ be the matrix consisting of rows $a'_{1 j_0}, a'_{2 j_0}, \ldots a'_{n j_0}$. Since $n = d$ and $a'_{i_0 j_0} = 0$ there exists $\beta \in ker(A') \setminus \{0\}$. Consider the following parameterizations:

$\lambda_{jl} = 0$	for $j > l$ and $l = j_0$
$\lambda_{jj} = 1$	for all $j \in [J]$
$\lambda_{jl} = 0$	else
$\vartheta_k = \beta$	for $k = j_0$
$\vartheta_k = 0$	else

Now we have that

$$\sum_{i=1}^{n} \sum_{j=1}^{J} \left(\sum_{k=1}^{j} \lambda_{j,j} \vartheta_k a_{ij} \right)^2 = \|\beta\|_2^2 > 0$$

and

$$\sum_{i=1}^{n}\sum_{j=1}^{J}\left(\sum_{k=1}^{j}\lambda_{j,j}\vartheta_k a'_{ij}\right)^2 = 0$$

thus $\{a'_{ij}\}_{i\in[n],j\in[J]}$ cannot be a coreset. □

B SENSITIVITY SAMPLING FRAMEWORK

The sensitivity sampling framework, as outlined and most recently updated in (Feldman et al., 2020), provides a methodology for generating coresets for optimization problems where the objective is to reduce the cost associated with and aggregate over the input data points. In this method, data points are selected at random, yet the selection probability for each point is proportional to its sensitivity with respect to the optimization problem, as an importance measure. This technique is designed to ensure the inclusion of pivotal points that might otherwise be missed under an unbiased uniform random selection due to the equally low probability of selection.

Definition B.1 (Sensitivity (Langberg and Schulman, 2010))**.** *Let $F = \{g_1, \dots, g_n\}$ be a family of functions that map from $\mathbb{R}^d$ to the non-negative real numbers, weighted with $w \in \mathbb{R}^n_{>0}$. The sensitivity of g_i for $f_w(\theta) = \sum_{j=1}^{n} w_j \cdot g(x_j\theta)$ is*

$$\zeta_i = \sup_{\theta\in\mathbb{R}^d, f_w(\theta)>0} \frac{w_i g_i(\theta)}{f_w(\theta)}.$$

The sum of the sensitivities $Z = \sum_{i=1}^{n}\zeta_i$ is called the total sensitivity.

The sensitivity sampling framework has demonstrated considerable advantages in the construction of coresets across various computational problems and statistical models, including linear regression (Clarkson, 2005; Drineas et al., 2006; Rudelson and Vershynin, 2007; Dasgupta et al., 2009), logistic regression (Munteanu et al., 2018), probit regression (Ding et al., 2024), and Poisson regression (Molina et al., 2018; Lie and Munteanu, 2024).

Sampling with probabilities proportional to sensitivity scores provably leads to a good approximation, although it requires the determination of the exact sensitivities to solve the original problem. However, it has been shown that the sample can also be drawn proportionally to any upper bounds such that $S = \sum_{i=1}^{n} s_i \geq \sum_{i=1}^{n}\zeta_i = Z$. Therefore, in order to be able to build a coreset using sensitivity sampling, it suffices to find the upper bounds to approximate a sensitivity. However, since the total sensitivity determines the sample size, this overestimation must be controlled carefully.

The following theorem builds the core for sensitivity sampling based coreset construction and is presented in its most recently updated and optimized version due to (Feldman et al., 2020).

Theorem B.2 ((Langberg and Schulman, 2010; Feldman and Langberg, 2011; Feldman et al., 2020))**.** *Let $F = \{g_1, \dots, g_n\}$ be a finite set of functions that map from $\mathbb{R}^d$ to $\mathbb{R}_{\geq 0}$ and let $w \in \mathbb{R}^n_{>0}$ be a vector of positive weights. Let ε, δ be in $(0, 1/2)$. Moreover, let $s_i \geq \zeta_i$ be upper bounds for the sensitivities and $S = \sum_{i=1}^{n} s_i \geq Z$. Then for given s_i, a set $R \subseteq F$ can be found in time $O(|F|)$ with*

$$|R| = O\left(\frac{S}{\varepsilon^2}\left(\Delta\log S + \log\left(\frac{1}{\delta}\right)\right)\right)$$

With the calculations of the weighted functions, such that with probability $1-\delta$ for all $\theta \in \mathbb{R}^d$ it holds:

$$(1-\varepsilon)\sum_{g\in F} w_i g_i(\theta) \leq \sum_{g\in R} u_i g_i(\theta) \leq (1+\varepsilon)\sum_{g\in F} w_i g_i(\theta)$$

Each element of R is drawn i.i.d. with probability $p_j = \frac{s_j}{S}$ from F, $u_i = \frac{Sw_j}{s_j|R|}$ denotes the weight of a function $g_i \in R$, which corresponds to $g_i \in F$, and Δ is an upper bound for the VC dimension of the Range Spaces $\mathcal{R}_{F^*}$, induced by F^* that we obtain by scaling all functions $g_i \in F$ with $\frac{Sw_j}{s_j|R|}$. It is thus

$$F^* = \left\{\frac{Sw_j}{s_j|R|}g_j(\theta) \;\middle|\; j \in [n]\right\}.$$

The set of functions $R \subseteq F$ with new weights u is a representation of our sought coreset. The size of R depends on both, the estimate of the total sensitivity, as well as the VC-dimension of the range spaces, which is induced by the reweighted function set F^*. Since the construction of the coreset is a probabilistic process, it cannot be ruled out that this may fail (unless we choose all data points for our coreset). The theorem thus introduces an controllable error probability δ, which also influences the size of the coreset logarithmically.

C ALGORITHMS

Algorithm 1: Hybrid coreset construction for MCTMs

Data: Full dataset $\mathcal{D} = \{y_i\}_{i=1}^n \subset \mathbb{R}^J$, target coreset size k
Result: Weighted coreset $\mathcal{C} = \{(y_j, w_j)\}_{j=1}^k$

Compute transformed statistics: Apply Bernstein polynomial transformation a of degree d to each y_i to obtain $B \in \mathbb{R}^{nJ \times dJ^2}$ as described in Section 2 resp. Section A.2
Compute ℓ_2 leverage scores of B: Use fast leverage score computation as in Theorem 2.13 of (Woodruff, 2014) to get a constant factor approximation for $u_i = \sup_{x \neq 0} \frac{|B_i x|^2}{\|Bx\|_2^2}$ for each $i \in [nJ]$.
Compute sensitivity proxy: For each i, set $s_i \leftarrow u_i + 1/n$ as a sensitivity score
Normalize to probabilities: $p_i \leftarrow \frac{s_i}{\sum_{j=1}^n s_j}$
Sampling phase:
(a) Let $k_1 = \lfloor \alpha k \rfloor$ (e.g. $\alpha = 0.8$) be the size of the sensitivity sample
(b) Sample k_1 points independently with probability $\{p_i\}$
(c) Assign weights $w_i \leftarrow \frac{1}{k_1 \cdot p_i}$ for each sampled point
Convex hull augmentation:
(a) Let $k_2 = k - k_1$
(b) Compute ε/J-kernel convex hull approximation as in (Blum et al., 2019) over the derivatives $\{a_j'(y_{ij})\}_{i \in [n], j \in [J]}$ and select k_2 extremal points
(c) Assign weight $w_j \leftarrow 1$ to each convex hull point
Form final coreset: Combine sampled points and hull points into $\mathcal{C}$ with associated weights w
Coreset fitting: Fit MCTM using weighted log-likelihood:

$$\hat{\theta}_{\text{coreset}} = \text{argmax}_\theta \sum_{(Y_j, w_j) \in \mathcal{C}} w_j \cdot \log f_\theta(Y_j)$$

D RELATIONSHIP BETWEEN MCTMS AND NORMALIZING FLOWS

Multivariate Conditional Transformation Models (MCTMs) and Normalizing Flows (NFs) are both powerful frameworks for modeling complex conditional probability distributions $p_Y(y \mid x)$. They are built upon the same mathematical foundation, namely the probability transformation formula, which enables density estimation by transforming a complex target distribution into a simpler base distribution through a bijective and differentiable function.

Let $Y \in \mathbb{R}^J$ be the response variable whose conditional density $p_Y(y \mid x)$ is of interest, and let $Z \in \mathbb{R}^J$ be a latent variable with a known base distribution $p_Z(z)$, typically a standard multivariate Gaussian. Both frameworks assume the existence of a transformation function $h : \mathbb{R}^J \times \mathcal{X} \to \mathbb{R}^J$ that is bijective in y for each fixed x, such that

$$Z = h(Y \mid x).$$

The inverse function $g = h^{-1}$ satisfies

$$Y = g(Z \mid x).$$

By the probability transformation formula, the conditional density of Y given x can be expressed as

$$p_Y(y \mid x) = p_Z(h(y \mid x)) \left| \det \left(\frac{\partial h(y \mid x)}{\partial y} \right) \right|.$$

Algorithm 2: Sparse approximation of the convex hull (Blum et al., 2019)

Data: A set of points P, a query point q, tolerance ε
Result: Approximated convex hull point t_M closest to q

Initialize first two points: randomly select one point a_0 and obtain a_1 as the furthest point to a_0
Obtain the third point a_2, which is the furthest point to the line $\overline{a_0 a_1}$. The set $\{a_0, a_1, a_2\}$ compose as the initial convex hull.
for $j \leftarrow 1$ **to** $n-3$ **do**
  $t_0 \leftarrow$ closest point of P to q
  $M \leftarrow O(1/\varepsilon^2)$
  for $i \leftarrow 1$ **to** M **do**
    $v_i \leftarrow q - t_{i-1}$
    $p_i \leftarrow$ point in P that is extremal in the direction of v_i
    if p_i *exists* **then**
      Compute the projection of q onto the line through t_{i-1} and p_i to find t_i
      $t_i \leftarrow$ the closest point to q on the line segment $s_i = t_{i-1}p_i$
    else
      $t_i \leftarrow t_{i-1}$
      break
    end
    if $\|q - t_i\| < \varepsilon$ *or* $i = M$ **then**
      return t_i
    end
  end
end

Taking the logarithm yields

$$\log p_Y(y \mid x) = \log p_Z(h(y \mid x)) + \log \left| \det \left(\frac{\partial h(y \mid x)}{\partial y} \right) \right|,$$

which is the objective typically maximized in both MCTM and NF during maximum likelihood estimation respectively training.

In MCTM, the transformation $h(y \mid x)$ is defined in a structured, semi-parametric way. It follows a lower triangular structure such that the j-th component $h_j(y \mid x)$ depends only on $y_1, \ldots, y_j$ and x. This triangular form ensures that the Jacobian matrix $\frac{\partial h(y|x)}{\partial y}$ is lower triangular. Each h_j is further decomposed as a linear combination of marginal transformations $\tilde{h}_\ell(y_\ell \mid x)$ for $\ell < j$, and its own marginal $\tilde{h}_j(y_j \mid x)$. Formally,

$$h_j(y_1, \ldots, y_j \mid x) = \sum_{\ell=1}^{j-1} \lambda_{j\ell}(x) \tilde{h}_\ell(y_\ell \mid x) + \tilde{h}_j(y_j \mid x).$$

In matrix-vector notation, the full transformation is written as $h(y \mid x) = \Lambda(x)\tilde{h}(y \mid x)$, where $\tilde{h}(y \mid x) = (\tilde{h}_1(y_1 \mid x), \ldots, \tilde{h}_J(y_J \mid x))^T$, and $\Lambda(x)$ is a lower triangular matrix with ones on the diagonal. Each marginal transformation $\tilde{h}_j$ is modeled using a basis function expansion of the form $\tilde{h}_j(y_j \mid x) = a_j(y_j)^T \vartheta_j(x)$, where $a_j(y_j)$ is a fixed basis vector and $\vartheta_j(x)$ is a parameter vector that may depend on x. Monotonicity is enforced by constraining the derivative $\frac{\partial \tilde{h}_j}{\partial y_j} > 0$. The Jacobian determinant simplifies significantly due to the triangular structure: it is the product of the marginal derivatives,

$$\det \left(\frac{\partial h(y \mid x)}{\partial y} \right) = \prod_{j=1}^{J} \frac{\partial \tilde{h}_j(y_j \mid x)}{\partial y_j} = \prod_{j=1}^{J} (a_j'(y_j))^T \vartheta_j(x).$$

Therefore, the conditional log-likelihood becomes

$$\log p_Y(y \mid x) = \log \phi_J(\Lambda(x)\tilde{h}(y \mid x)) + \sum_{j=1}^{J} \log((a_j'(y_j))^T \vartheta_j(x)),$$

where ϕ_J denotes the density of the standard J-variate normal distribution.

Normalizing Flows employ the same principle of density transformation, but define $h(y \mid x)$ or its inverse $g(z \mid x)$ through a sequence of invertible mappings composed of simple building blocks. These mappings are typically parameterized using deep neural networks. A transformation in NF is written as $h = h_L \circ \cdots \circ h_1$, where each h_ℓ is a bijective transformation with tractable Jacobian determinant. The parameters of each transformation may be functions of x, enabling conditional modeling.

Popular designs include autoregressive flows, where each output h_j depends only on $y_1, \ldots, y_{j-1}$ and x, resulting in a lower triangular Jacobian similar to MCTMs. Another class of NF architectures is based on coupling layers, where the input vector is split into two parts, one of which remains fixed while the other is transformed using an affine function whose parameters are learned from the fixed part and the context x. These transformations are composed repeatedly, with role reversal between parts in successive layers to ensure expressiveness.

Despite their differences in parameterization and implementation, MCTM and Normalizing Flows share the same mathematical foundation and ultimately rely on the same probability transformation principle. Both frameworks define a transformation from the observed data space to a latent space with a known density, compute the associated Jacobian determinant, and maximize the resulting log-likelihood over the data.

E EXPERIMENTAL DESIGN

E.1 Simulation Study

E.1.1 Data Generation Processes

To thoroughly evaluate our proposed method across diverse dependency structures, we implemented 14 different data generation processes. Each process was designed to test specific aspects of dependency modeling. For all processes, we generated datasets with $n = 10\,000$ samples and evaluated coreset performance at sizes 30 and 100. Below, we describe each process mathematically.

1. Bivariate Normal Distribution

The standard bivariate normal distribution with correlation parameter ρ:

$$(Y_1, Y_2) \sim \mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & \rho\\ \rho & 1\end{bmatrix}\right)$$

where $\rho = 0.7$ represents the correlation between variables. This baseline case tests performance on linear dependency structures.

2. Non-linear Correlation

A non-linear correlation structure where the correlation coefficient varies with the predictor:

$$\begin{aligned} &Y_1 = X^2 + \varepsilon_1, \quad \varepsilon_1 \sim \mathcal{N}(0, 0.5^2)\\ &Y_2 \sim \mathcal{N}(0,1), \text{ with correlation } \rho(X) = \sin(X) \text{ to } Y_1 \end{aligned}$$

where $X \in [-3, 3]$. This tests the ability to capture location-dependent correlation.

3. Bivariate Normal Mixture

A mixture of two bivariate normal distributions with different means and correlation structures:

$$\begin{aligned} (Y_1, Y_2) \sim\ & 0.5 \cdot \mathcal{N}\left(\begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}1 & 0.8\\ 0.8 & 1\end{bmatrix}\right)\\ & + 0.5 \cdot \mathcal{N}\left(\begin{bmatrix}3\\-2\end{bmatrix}, \begin{bmatrix}1.5 & -0.5\\ -0.5 & 1.5\end{bmatrix}\right) \end{aligned}$$

This tests performance on multimodal data with different local dependency structures.

4. Geometric Mixed Distribution

A distribution combining two distinct geometric patterns:

$$(Y_1, Y_2) \sim 0.5 \cdot \text{Circular} + 0.5 \cdot \text{Cross}$$

where the Circular component generates points along a circle with radius $r \sim \mathcal{N}(2, 0.2^2)$ and uniform angle $\theta \sim \text{Uniform}(0, 2\pi)$, while the Cross component generates points along two perpendicular lines with controlled variance. This distribution tests the ability to capture multiple geometric structures simultaneously, with continuous circular features intersecting with linear patterns. The sharp differences in geometric structure and local data density create a particularly challenging scenario for coreset selection.

5. Skewed t-Distribution

A heavy-tailed distribution with skewness:

$$(Y_1, Y_2) \sim \text{Skew-}t_\nu\left(\xi, \Omega, \alpha\right)$$

where $\xi = [0, 0]^T$ is the location parameter, $\Omega = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$ is the scale matrix, $\alpha = [5, -3]^T$ is the skewness parameter, and $\nu = 4$ specifies the degrees of freedom. This tests performance on asymmetric, heavy-tailed distributions.

6. Heteroscedastic Distribution

A distribution where the variance depends on the location:

$$\begin{aligned} Y_1 &\sim \mathcal{N}(X^2, \sigma_1^2(X)) \text{ where } \sigma_1(X) = e^{0.5X} \\ Y_2 &\sim \mathcal{N}(\sin(X), \sigma_2^2(X)) \text{ where } \sigma_2(X) = \sqrt{|X|} \end{aligned}$$

with $X \in [-3, 3]$. This tests the ability to capture variance heterogeneity.

7. Copula Complex Distribution

A Clayton copula-based dependency structure with gamma and log-normal marginals:

$$\begin{aligned} (U_1, U_2) &\sim C_{\text{Clayton}}(\theta = 2) \\ Y_1 &= F^{-1}_{\text{Gamma}(2,1)}(U_1) \\ Y_2 &= F^{-1}_{\text{LogNormal}(0,1)}(U_2) \end{aligned}$$

This tests performance on complex dependency structures with non-normal marginals.

8. Spiral Dependency

A spiral pattern with added noise:

$$\begin{aligned} t &\in [0, 3\pi] \\ r &= 0.5t \\ Y_1 &= r\cos(t) + \varepsilon_1, \quad \varepsilon_1 \sim \mathcal{N}(0, 0.5^2) \\ Y_2 &= r\sin(t) + \varepsilon_2, \quad \varepsilon_2 \sim \mathcal{N}(0, 0.5^2) \end{aligned}$$

This tests performance on complex geometric dependencies.

9. Circular Dependency

A circular pattern with radius variation:

$$\begin{aligned} \theta &\sim \text{Uniform}(0, 2\pi) \\ r &\sim \mathcal{N}(5, 1^2) \\ Y_1 &= r\cos(\theta) \\ Y_2 &= r\sin(\theta) \end{aligned}$$

This tests the ability to capture circular dependencies where linear correlation measures fail.

10. t-Copula Dependency

A t-copula with t and exponential marginals:

$$\begin{aligned}(U_1, U_2) &\sim C_t(\rho = 0.7, \nu = 3)\\ Y_1 &= F_{t_5}^{-1}(U_1)\\ Y_2 &= F_{\text{Exp}(1)}^{-1}(U_2)\end{aligned}$$

This tests performance on tail dependencies with asymmetric marginals.

11. Piecewise Dependency

A piecewise function with different correlation regimes:

$$\begin{aligned}Y_1 &\sim \mathcal{N}(0, 2^2)\\ Y_2 &= \begin{cases}1.5Y_1 + \varepsilon_1, & \text{if } Y_1 < -1\\ -0.5Y_1 + \varepsilon_2, & \text{if } -1 \le Y_1 < 1\\ -2Y_1 + \varepsilon_3, & \text{if } Y_1 \ge 1\end{cases}\end{aligned}$$

where $\varepsilon_1, \varepsilon_3 \sim \mathcal{N}(0, 0.5^2)$ and $\varepsilon_2 \sim \mathcal{N}(0, 0.8^2)$. This tests ability to capture regime-dependent correlations.

12. Hourglass Dependency

A heteroscedastic pattern with variance increasing quadratically from center:

$$\begin{aligned}Y_1 &\sim \mathcal{N}(0, 2^2)\\ Y_2 &\sim \mathcal{N}(0, \sigma^2(Y_1)) \text{ where } \sigma^2(Y_1) = 0.2 + 0.3Y_1^2\end{aligned}$$

This tests ability to capture complex variance structures.

13. Bimodal Clusters

Two distinct clusters with opposing correlation structures:

$$\begin{aligned}(Y_1, Y_2) \sim\ & 0.5 \cdot \mathcal{N}\left(\begin{bmatrix}-2\\ 2\end{bmatrix}, \begin{bmatrix}1 & 0.8\\ 0.8 & 1\end{bmatrix}\right)\\ & + 0.5 \cdot \mathcal{N}\left(\begin{bmatrix}2\\ 2\end{bmatrix}, \begin{bmatrix}1 & -0.7\\ -0.7 & 1\end{bmatrix}\right)\end{aligned}$$

This tests the ability to capture cluster-specific dependency structures.

14. Sinusoidal Dependency

A sinusoidal relationship with noise:

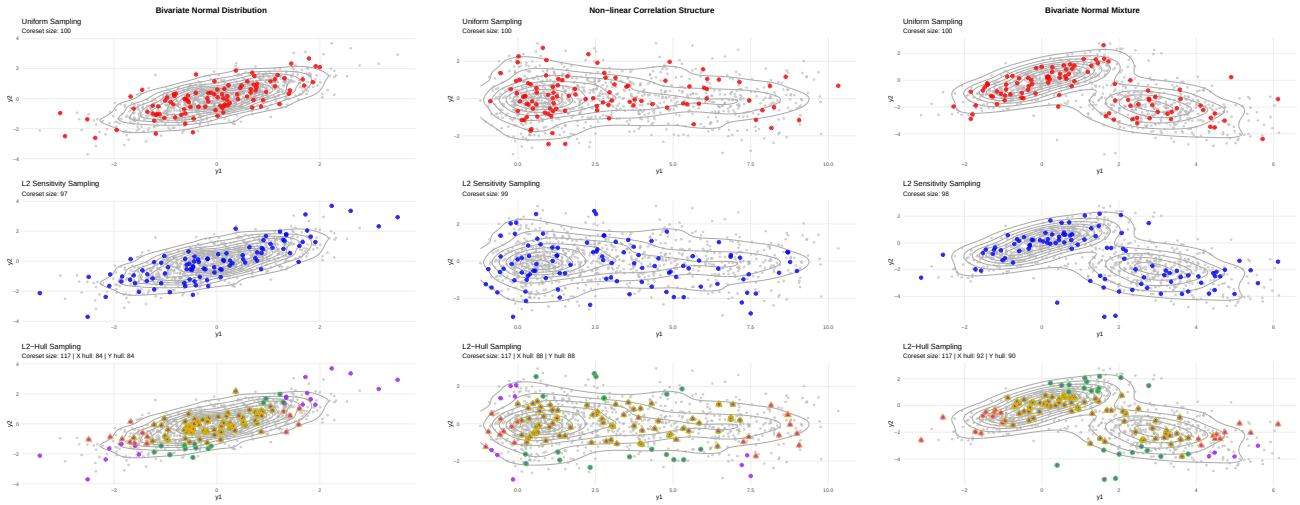
$$\begin{aligned}Y_1 &\in [-3, 3]\\ Y_2 &= 2\sin(\pi Y_1) + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, 0.5^2)\end{aligned}$$

This tests performance on periodic dependencies.

These 14 data generation processes provide a comprehensive evaluation framework for testing our coreset method across a wide range of dependency structures, from simple linear correlations to complex geometric patterns and regime-dependent relationships.

E.1.2 Simulation Data Visualization

We visualize each data generation process with coresets created using three different sampling methods: Uniform Sampling, ℓ_2 Sensitivity Sampling, and our proposed ℓ_2-Hull Sampling method. Each visualization shows how well the different coreset methods preserve the underlying data distribution structure.

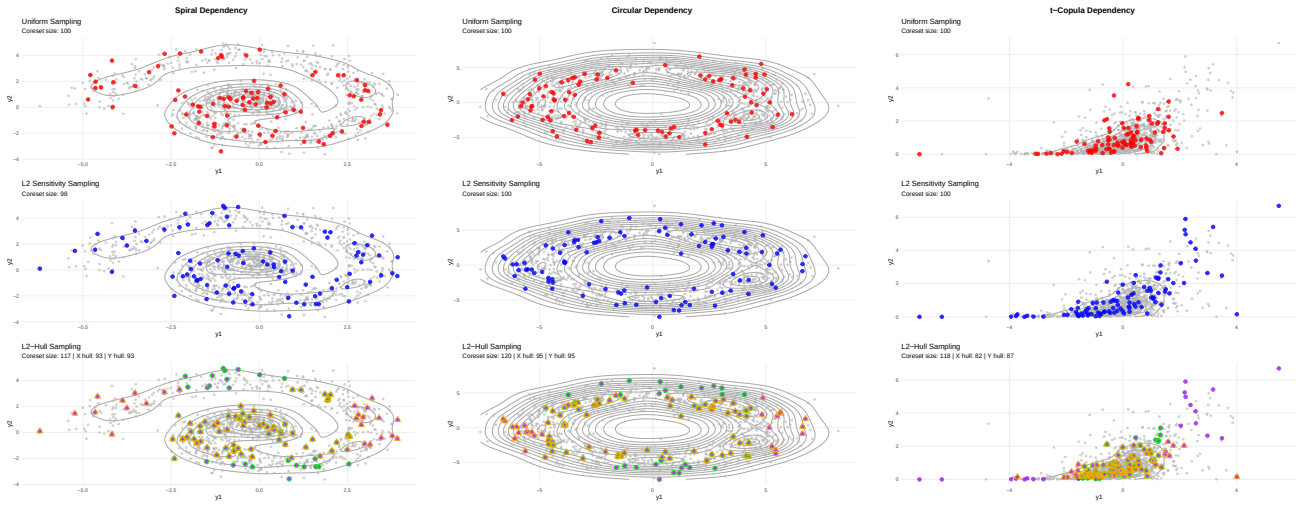


(a) Bivariate Normal (b) Non-linear Correlation (c) Bivariate Normal Mixture

Figure 2: Coreset visualization for basic probability distributions. Each row shows a different sampling method: Uniform (top), ℓ_2 Sensitivity (middle), and ℓ_2-Hull (bottom).

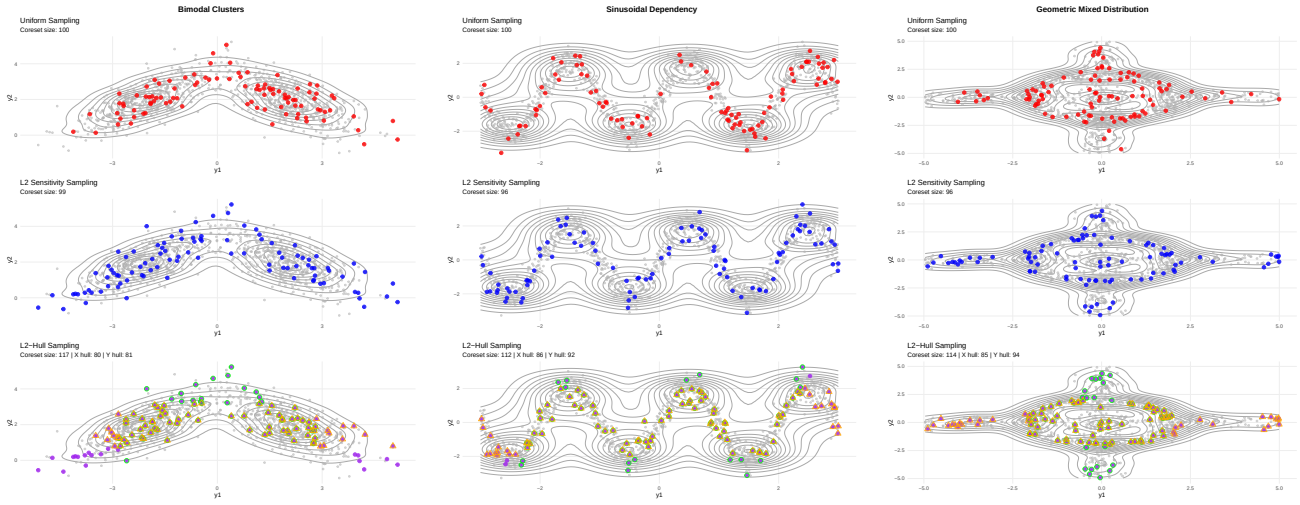
(a) Skew-t Distribution (b) Heteroscedastic Distribution (c) Copula Complex Distribution

Figure 3: Coreset visualization for complex probability distributions. Each column shows a different sampling method: Uniform (top), ℓ_2 Sensitivity (middle), and ℓ_2-Hull (bottom).



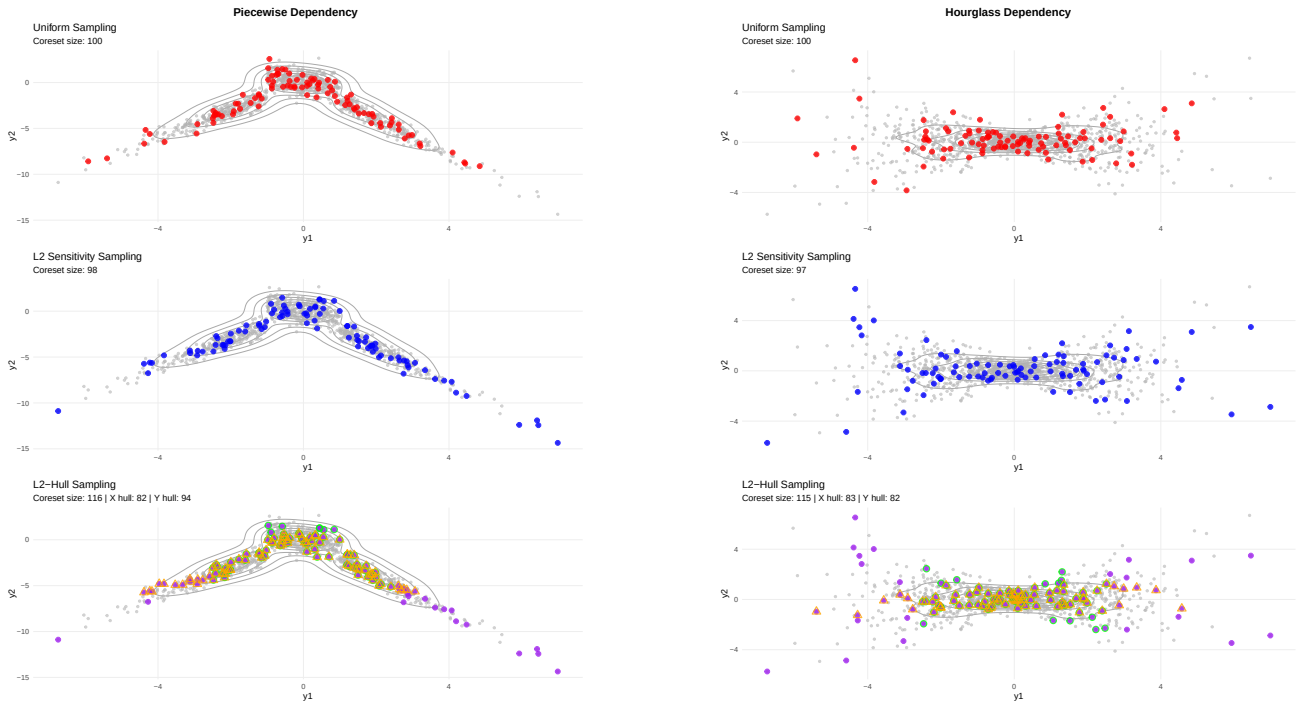
(a) Spiral Dependency (b) Circular Dependency (c) t-Copula Dependency

Figure 4: Coreset visualization for geometric dependency structures. Each column shows a different sampling method: Uniform (top), ℓ_2 Sensitivity (middle), and ℓ_2-Hull (bottom).



(a) Bimodal Clusters (b) Sinusoidal Dependency (c) Mixture Distribution

Figure 5: Coreset visualization for additional dependency structures. Each column shows a different sampling method: Uniform (top), ℓ_2 Sensitivity (middle), and ℓ_2-Hull (bottom).



(a) Piecewise Dependency

(b) Hourglass Dependency

Figure 6: Coreset visualization for functional dependency structures. Each column shows a different sampling method: Uniform (top), ℓ_2 Sensitivity (middle), and ℓ_2-Hull (bottom).

To visually demonstrate the advantages of our proposed method, we present coreset visualizations for representative data generation processes. In each figure, we display coresets of approximately 100 points generated from original samples of 1 000 points using three different sampling methods: Uniform Sampling (top), ℓ_2 Sensitivity Sampling (middle), and our proposed ℓ_2-Hull Sampling (bottom).

As shown in Figures 2–5, our ℓ_2-Hull method effectively captures the entire shape of the underlying data distribution, particularly in complex scenarios such as the Hourglass Dependency, Piecewise Dependency, Spiral Dependency, Copula Complex Distribution, and Bimodal Clusters. The visualization highlights a key limitation of uniform sampling: it often fails to include points that are distant from the central mass of the distribution but are nevertheless critical for accurately representing the overall data structure.

For instance, in the Bimodal Clusters visualization (Figure 5a), our ℓ_2-Hull method ensures comprehensive coverage of both clusters including their boundaries, while uniform sampling misses several critical points that define the extent of the distribution. Similarly, in the Piecewise Dependency case (Figure 6a), the non-linear relationship is better preserved by our method, particularly at the extremes of the distribution.

It is worth noting that in many scenarios, the performances of ℓ_2 Sensitivity Sampling and our ℓ_2-Hull method appear similar, especially as the coreset size increases. This is expected, as the primary purpose of adding convex hull points is to safeguard against extreme cases that may arise during optimization. As the sample size grows, the probability of encountering such extreme cases diminishes. Nevertheless, the ℓ_2-Hull method provides a theoretical guarantee that important boundary points are always included, regardless of the particular dataset instance.

The enhanced coverage provided by our method translates directly to the improved empirical performance metrics observed in Tables 3 and 4, particularly for complex dependency structures where capturing the overall shape of the distribution is crucial for accurate statistical inference.

E.1.3 Simulation Experiments Results

In all the above scenarios, we will evaluate under the same sampling/modeling process and can adjust the sample size n as well as the correlation parameters (e.g., ρ) as needed.

Coreset Construction Methods We compare three sampling strategies:

1. **Uniform Subsampling**: k points are taken with equal probability from all n samples without replacement, and the weights are set to $\frac{n}{k}$.
2. ℓ_2 **-Only Leverage Score Sampling**: leverage scores (or upper bounds on sensitivity) are calculated for $a(x_i)$ and $a(y_i)$, respectively, and subsamples are combined and reweighted after sampling them with the probability $p_i \propto \ell_2$ leverage score; and the weight is then determined using *merge probability* to compute the final weights and do the normalization.
3. ℓ_2**-Hull Hybrid**: on the basis of the above ℓ_2 sensitivity sampling, we also perform *convex hull* (or ε-kernel) approximation sampling on its derivative matrix $a'(\cdot)$ by adding an extra batch of points located at the extreme geometric boundaries, thus avoiding the logarithmic term $\log(a'(x_i)^T\vartheta)$ in the likelihood function on which the values are unstable or near zero. The *sensitivity subsample* and *convex hull subsample* are eventually summarized together in a joint index.

Main Workflow For each data generation method, we follow the following flow of experiments:

1. **Data Generation:** Call the corresponding function (e.g. `generate_mixture`) to generate the samples, which can be randomly initialized multiple times.
2. **Full Data Baseline:** Perform MCTM fitting with full data, record its log-likelihood ℓ_{full} with the estimated coefficients $\hat{\theta}_{full}$, as a baseline.
3. **Coreset Sampling & Fitting:** Use *Uniform Sampling*, ℓ_2*-Only*, and ℓ_2*-Hull*, respectively, for multiple subset sizes from the set $k \in \{\text{min\_size}, \ldots, \text{max\_size}\}$ to obtain a subsample and compute the weights; subsequently, the MCTM is fitted using only this subsample to obtain the log-likelihood $\ell_{coreset}$ and estimated coefficients $\hat{\theta}_{coreset}$.
4. **Evaluation Metrics:**
   - **Likelihood Ratio:** compare $\ell_{coreset}/\ell_{full}$, if it is close to 1;
   - **Parameter Error:** calculate the value of $\|\hat{\vartheta}_{coreset} - \hat{\vartheta}_{full}\|_2^2$ and convergence trend under different k;
   - **Lambda Error:** Measure the absolute difference in the dependency parameter $|\lambda_{coreset} - \lambda_{full}|$ to evaluate how well the coreset preserves the dependency structure;
   - **Time Cost:** Record the sampling time consumption and optimization time consumption respectively for evaluating the efficiency of the algorithm.

Summary With such multi-scenario simulations, we can systematically examine the proposed MCTM coreset scheme under different distributions. Detailed comparative graphs and numerical analyses showing the combined advantages of the new method in terms of likelihood approximation accuracy, parameter estimation bias, and time efficiency are presented in the experimental results below.

Figure 7 visualizes the performance specifically for three distributions: bivariate normal mixture, non-linear correlation, and bimodal cluster distribution. In Figure 7, the red line corresponds to our proposed method combining ℓ_2 subsampling with convex hull approximation, the blue line denotes the ℓ_2 subsampling method alone, and the green line indicates the traditional uniform subsampling method. All simulations are based on an original dataset size of 10 000 points.

To further illustrate the advantages of our coreset approach, we show the effect of reconstructing the univariate marginal densities after constructing the coreset with three different sampling strategies on simulated data from a binary normal distribution, with realistic marginal distribution predictions for X and Y in Figure 10 and

Table 3: Performance comparison of different coreset methods for various data generation processes (coreset Size = 30)

Data Gen. Process	Method	Param. ℓ_2 dist.	λ error	Likelihood ratio	Rel. Impr.(%)	Total time(s)
Bivariate normal	ℓ_2-hull	2.56 ± 0.76	0.44 ± 0.16	1.54 ± 0.29	12.8	0.21 ± 0.03
	ℓ_2-only	$\mathbf{2.54 \pm 0.62}$	0.51 ± 0.13	1.65 ± 0.33	1.6	0.20 ± 0.02
	uniform	4.91 ± 4.74	$\mathbf{0.29 \pm 0.26}$	1.94 ± 1.13	baseline	$\mathbf{0.13 \pm 0.03}$
Non-linear correlation	ℓ_2-hull	$\mathbf{1.76 \pm 0.88}$	$\mathbf{0.09 \pm 0.09}$	$\mathbf{1.03 \pm 0.01}$	**49.8**	0.19 ± 0.02
	ℓ_2-only	2.18 ± 0.82	0.10 ± 0.10	1.05 ± 0.02	38.8	0.19 ± 0.01
	uniform	3.08 ± 0.91	0.11 ± 0.07	1.21 ± 0.46	baseline	$\mathbf{0.12 \pm 0.01}$
Bivariate normal mixture	ℓ_2-hull	$\mathbf{2.60 \pm 1.00}$	$\mathbf{0.14 \pm 0.07}$	$\mathbf{1.07 \pm 0.03}$	**49.6**	0.20 ± 0.03
	ℓ_2-only	2.76 ± 0.94	0.16 ± 0.10	1.08 ± 0.05	43.7	0.21 ± 0.04
	uniform	4.14 ± 1.80	0.20 ± 0.15	1.42 ± 0.34	baseline	$\mathbf{0.15 \pm 0.04}$
Geometric Mixed Distribution	ℓ_2-hull	$\mathbf{2.51 \pm 2.61}$	$\mathbf{0.06 \pm 0.04}$	$\mathbf{1.33 \pm 0.53}$	**41.4**	0.20 ± 0.04
	ℓ_2-only	4.09 ± 4.35	0.07 ± 0.03	1.58 ± 0.58	9.0	0.20 ± 0.02
	uniform	4.11 ± 2.49	0.14 ± 0.09	1.47 ± 0.53	baseline	0.13 ± 0.03
Skew-t distribution	ℓ_2-hull	$\mathbf{3.55 \pm 1.46}$	0.61 ± 0.27	$\mathbf{1.90 \pm 0.70}$	0	0.23 ± 0.03
	ℓ_2-only	3.72 ± 1.36	0.70 ± 0.26	2.14 ± 0.91	0	0.23 ± 0.09
	uniform	4.14 ± 2.04	$\mathbf{0.36 \pm 0.31}$	2.17 ± 0.96	baseline	$\mathbf{0.15 \pm 0.07}$
Heteroscedastic distribution	ℓ_2-hull	$\mathbf{2.07 \pm 0.69}$	$\mathbf{0.08 \pm 0.05}$	$\mathbf{1.07 \pm 0.12}$	**64.8**	0.20 ± 0.07
	ℓ_2-only	2.65 ± 0.88	$\mathbf{0.08 \pm 0.05}$	1.09 ± 0.13	58.4	0.20 ± 0.02
	uniform	4.47 ± 3.38	0.16 ± 0.14	1.63 ± 1.19	baseline	$\mathbf{0.13 \pm 0.02}$
Copula complex distribution	ℓ_2-hull	$\mathbf{4.56 \pm 1.57}$	$\mathbf{0.18 \pm 0.13}$	$\mathbf{1.23 \pm 0.20}$	**58.0**	0.26 ± 0.04
	ℓ_2-only	5.30 ± 1.48	0.20 ± 0.15	1.36 ± 0.29	51.1	0.28 ± 0.08
	uniform	11.05 ± 7.80	0.27 ± 0.22	2.35 ± 0.56	baseline	$\mathbf{0.18 \pm 0.06}$
Spiral dependency	ℓ_2-hull	$\mathbf{1.86 \pm 0.55}$	$\mathbf{0.04 \pm 0.03}$	$\mathbf{1.04 \pm 0.01}$	**79.5**	0.21 ± 0.03
	ℓ_2-only	2.38 ± 0.80	0.06 ± 0.05	1.07 ± 0.02	71.9	0.22 ± 0.07
	uniform	6.16 ± 3.50	0.15 ± 0.09	2.05 ± 0.60	baseline	$\mathbf{0.21 \pm 0.22}$
Circular dependency	ℓ_2-hull	$\mathbf{1.51 \pm 0.39}$	$\mathbf{0.06 \pm 0.04}$	$\mathbf{1.02 \pm 0.01}$	**59.3**	0.20 ± 0.02
	ℓ_2-only	1.95 ± 0.62	$\mathbf{0.06 \pm 0.04}$	1.12 ± 0.29	46.3	0.21 ± 0.03
	uniform	4.11 ± 3.24	0.08 ± 0.08	1.33 ± 0.73	baseline	$\mathbf{0.14 \pm 0.03}$
t Copula	ℓ_2-hull	$\mathbf{5.77 \pm 1.61}$	0.51 ± 0.30	$\mathbf{1.87 \pm 0.62}$	0	0.26 ± 0.03
	ℓ_2-only	7.18 ± 1.63	0.60 ± 0.29	1.98 ± 0.63	0	0.24 ± 0.03
	uniform	6.58 ± 3.35	$\mathbf{0.23 \pm 0.22}$	2.56 ± 0.92	baseline	$\mathbf{0.23 \pm 0.25}$
Piecewise dependency	ℓ_2-hull	$\mathbf{2.20 \pm 0.99}$	$\mathbf{0.30 \pm 0.13}$	$\mathbf{1.11 \pm 0.19}$	**58.5**	0.21 ± 0.04
	ℓ_2-only	2.38 ± 0.83	0.35 ± 0.15	$\mathbf{1.11 \pm 0.15}$	53.4	0.19 ± 0.02
	uniform	5.48 ± 4.11	0.46 ± 0.41	1.53 ± 0.74	baseline	$\mathbf{0.13 \pm 0.03}$
Hourglass dependency	ℓ_2-hull	1.99 ± 1.19	$\mathbf{0.14 \pm 0.07}$	$\mathbf{1.04 \pm 0.02}$	**55.8**	0.19 ± 0.03
	ℓ_2-only	$\mathbf{1.85 \pm 1.04}$	0.15 ± 0.08	1.06 ± 0.03	51.7	0.18 ± 0.01
	uniform	5.00 ± 3.21	0.16 ± 0.15	1.65 ± 0.69	baseline	$\mathbf{0.12 \pm 0.01}$
Bimodal clusters	ℓ_2-hull	$\mathbf{2.14 \pm 0.47}$	$\mathbf{0.15 \pm 0.09}$	$\mathbf{1.04 \pm 0.01}$	**58.5**	0.19 ± 0.02
	ℓ_2-only	2.61 ± 0.66	0.17 ± 0.09	1.06 ± 0.02	51.7	0.18 ± 0.02
	uniform	4.43 ± 3.05	0.22 ± 0.16	1.61 ± 0.74	baseline	$\mathbf{0.14 \pm 0.04}$
Sinusoidal dependency	ℓ_2-hull	$\mathbf{1.42 \pm 0.48}$	$\mathbf{0.05 \pm 0.05}$	$\mathbf{1.02 \pm 0.01}$	**38.6**	0.18 ± 0.03
	ℓ_2-only	1.66 ± 0.50	0.09 ± 0.08	1.03 ± 0.01	16.8	0.19 ± 0.02
	uniform	1.91 ± 0.74	0.10 ± 0.06	1.04 ± 0.01	baseline	$\mathbf{0.12 \pm 0.01}$

Note: The results in the table are the mean ± 1 standard deviation of independent simulations. The Relative Improvement is calculated as the average percentage improvement across all metrics, where improvement for parameter ℓ_2 distance and λ error is (baseline − method)/baseline × 100%, and for likelihood ratio it is $(|\text{baseline} - 1| - |\text{method} - 1|)/|\text{baseline} - 1| \times 100\%$. The parameters ℓ_2 distance and λ error are better when smaller, and the log-likelihood ratio is better when closer to 1. Negative relative improvements are shown as 0. The best performance for each metric in each data generation process is highlighted in **bold**.

Table 4: Performance comparison of different coreset methods for various data generation processes (coreset Size = 100)

Data Gen. Process	Method	Param. ℓ_2 dist.	λ error	Likelihood ratio	Rel. Impr.(%)	Total time(s)
Bivariate normal	ℓ_2-hull	$\mathbf{1.80 \pm 0.42}$	0.30 ± 0.07	1.32 ± 0.11	0	0.23 ± 0.02
	ℓ_2-only	2.01 ± 0.36	0.39 ± 0.09	1.48 ± 0.17	0	0.24 ± 0.04
	uniform	1.98 ± 0.66	$\mathbf{0.11 \pm 0.06}$	$\mathbf{1.19 \pm 0.08}$	baseline	$\mathbf{0.13 \pm 0.02}$
Non-linear correlation	ℓ_2-hull	$\mathbf{1.05 \pm 0.42}$	$\mathbf{0.03 \pm 0.03}$	$\mathbf{1.01 \pm 0.00}$	**52.6**	0.20 ± 0.02
	ℓ_2-only	1.20 ± 0.53	$\mathbf{0.03 \pm 0.03}$	$\mathbf{1.01 \pm 0.01}$	38.8	0.22 ± 0.02
	uniform	1.79 ± 0.79	0.08 ± 0.05	1.02 ± 0.01	baseline	$\mathbf{0.11 \pm 0.02}$
Bivariate normal mixture	ℓ_2-hull	1.54 ± 0.42	$\mathbf{0.04 \pm 0.04}$	$\mathbf{1.06 \pm 0.01}$	**34.6**	0.30 ± 0.22
	ℓ_2-only	$\mathbf{1.53 \pm 0.45}$	0.07 ± 0.04	$\mathbf{1.06 \pm 0.02}$	26.4	0.21 ± 0.02
	uniform	1.99 ± 0.58	0.09 ± 0.09	1.09 ± 0.04	baseline	$\mathbf{0.13 \pm 0.02}$
Geometric Mixed Distribution	ℓ_2-hull	$\mathbf{1.19 \pm 0.51}$	$\mathbf{0.02 \pm 0.02}$	$\mathbf{1.01 \pm 0.00}$	**49.4**	0.20 ± 0.02
	ℓ_2-only	1.27 ± 0.60	0.03 ± 0.02	1.01 ± 0.00	42.1	0.21 ± 0.03
	uniform	1.75 ± 0.66	0.06 ± 0.04	1.02 ± 0.01	baseline	0.13 ± 0.03
Skew-t distribution	ℓ_2-hull	$\mathbf{2.04 \pm 0.42}$	0.26 ± 0.14	$\mathbf{1.24 \pm 0.10}$	8.1	0.22 ± 0.01
	ℓ_2-only	$\mathbf{1.99 \pm 0.33}$	0.33 ± 0.16	1.28 ± 0.12	0	0.22 ± 0.03
	uniform	2.67 ± 1.11	$\mathbf{0.20 \pm 0.15}$	1.34 ± 0.25	baseline	$\mathbf{0.12 \pm 0.02}$
Heteroscedastic distribution	ℓ_2-hull	$\mathbf{1.06 \pm 0.27}$	$\mathbf{0.04 \pm 0.03}$	$\mathbf{1.01 \pm 0.00}$	**27.6**	0.24 ± 0.08
	ℓ_2-only	1.34 ± 0.40	$\mathbf{0.04 \pm 0.02}$	1.02 ± 0.00	11.3	0.21 ± 0.02
	uniform	1.46 ± 0.84	0.09 ± 0.05	1.01 ± 0.01	baseline	$\mathbf{0.13 \pm 0.02}$
Copula complex distribution	ℓ_2-hull	$\mathbf{3.53 \pm 0.70}$	$\mathbf{0.12 \pm 0.09}$	$\mathbf{1.15 \pm 0.10}$	**32.0**	0.26 ± 0.04
	ℓ_2-only	4.05 ± 0.78	0.13 ± 0.08	1.16 ± 0.08	24.1	0.26 ± 0.03
	uniform	3.75 ± 1.42	0.18 ± 0.15	1.34 ± 0.27	baseline	$\mathbf{0.18 \pm 0.04}$
Spiral dependency	ℓ_2-hull	$\mathbf{1.37 \pm 0.43}$	$\mathbf{0.02 \pm 0.02}$	$\mathbf{1.01 \pm 0.01}$	**34.4**	0.25 ± 0.08
	ℓ_2-only	1.52 ± 0.37	0.03 ± 0.02	1.03 ± 0.01	0	0.27 ± 0.19
	uniform	1.88 ± 0.57	0.04 ± 0.03	1.02 ± 0.01	baseline	$\mathbf{0.13 \pm 0.01}$
Circular dependency	ℓ_2-hull	$\mathbf{0.94 \pm 0.36}$	$\mathbf{0.02 \pm 0.01}$	$\mathbf{1.01 \pm 0.00}$	**54.2**	0.21 ± 0.01
	ℓ_2-only	1.05 ± 0.20	0.03 ± 0.02	$\mathbf{1.01 \pm 0.00}$	43.9	0.21 ± 0.01
	uniform	1.83 ± 0.66	0.05 ± 0.04	1.01 ± 0.01	baseline	$\mathbf{0.12 \pm 0.01}$
t Copula	ℓ_2-hull	3.86 ± 1.03	0.28 ± 0.25	1.39 ± 0.38	0	0.27 ± 0.04
	ℓ_2-only	4.71 ± 1.26	0.32 ± 0.27	1.49 ± 0.50	0	0.28 ± 0.15
	uniform	$\mathbf{2.43 \pm 1.04}$	$\mathbf{0.13 \pm 0.14}$	$\mathbf{1.16 \pm 0.16}$	baseline	$\mathbf{0.19 \pm 0.11}$
Piecewise dependency	ℓ_2-hull	$\mathbf{1.06 \pm 0.36}$	$\mathbf{0.09 \pm 0.08}$	$\mathbf{1.01 \pm 0.01}$	**49.8**	0.22 ± 0.03
	ℓ_2-only	1.29 ± 0.27	0.12 ± 0.11	$\mathbf{1.01 \pm 0.01}$	29.8	0.21 ± 0.02
	uniform	1.62 ± 0.90	0.15 ± 0.16	1.03 ± 0.04	baseline	$\mathbf{0.12 \pm 0.02}$
Hourglass dependency	ℓ_2-hull	$\mathbf{0.96 \pm 0.33}$	$\mathbf{0.08 \pm 0.09}$	$\mathbf{1.01 \pm 0.00}$	**46.0**	0.21 ± 0.02
	ℓ_2-only	0.97 ± 0.40	0.09 ± 0.09	1.02 ± 0.01	38.6	0.20 ± 0.01
	uniform	1.72 ± 0.54	0.12 ± 0.09	1.04 ± 0.02	baseline	$\mathbf{0.11 \pm 0.01}$
Bimodal clusters	ℓ_2-hull	$\mathbf{0.95 \pm 0.24}$	$\mathbf{0.05 \pm 0.05}$	$\mathbf{1.01 \pm 0.00}$	**46.5**	0.20 ± 0.01
	ℓ_2-only	1.03 ± 0.32	0.06 ± 0.04	1.02 ± 0.00	34.9	0.20 ± 0.01
	uniform	1.63 ± 0.53	0.11 ± 0.10	1.02 ± 0.01	baseline	$\mathbf{0.12 \pm 0.01}$
Sinusoidal dependency	ℓ_2-hull	$\mathbf{1.24 \pm 0.39}$	$\mathbf{0.03 \pm 0.03}$	$\mathbf{1.01 \pm 0.00}$	**42.0**	0.20 ± 0.01
	ℓ_2-only	1.28 ± 0.44	0.07 ± 0.04	$\mathbf{1.01 \pm 0.01}$	15.6	0.21 ± 0.01
	uniform	1.36 ± 0.40	0.07 ± 0.04	1.02 ± 0.01	baseline	$\mathbf{0.11 \pm 0.01}$

Note: The results in the table are the mean ± 1 standard deviation of independent simulations. The Relative Improvement is calculated as the average percentage improvement across all metrics, where improvement for parameter ℓ_2 distance and λ error is (baseline − method)/baseline × 100%, and for likelihood ratio it is $(|\text{baseline} - 1| - |\text{method} - 1|)/|\text{baseline} - 1| \times 100\%$. The parameters ℓ_2 distance and λ error are better when smaller, and the log-likelihood ratio is better when closer to 1. Negative relative improvements are shown as 0. The best performance for each metric in each data generation process is highlighted in **bold**.

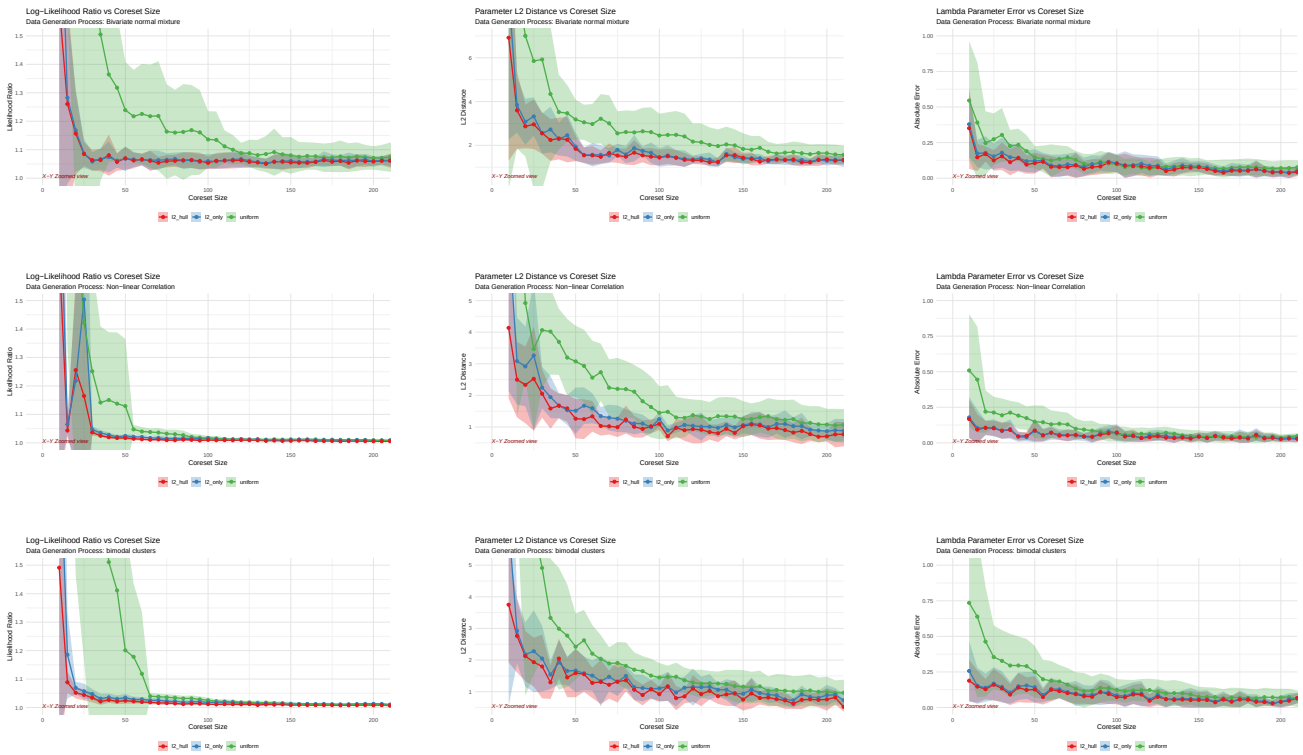


Figure 7: Convergence of the likelihood ratio, parameter error, and λ error as coreset size increases. First row: Bivariate mixture data of two Gaussian distributions with different means and variances. Second row: Non-linear correlation. Third row: Bimodal clusters with two distinct clusters with opposing correlation structure.

Figure 11, respectively. Each column corresponds to a different coreset size $k = 50, 100, 500$, while the three rows show uniform sampling, ℓ_2 sampling, and ℓ_2-hull sampling, respectively. To reflect the stability of the methods, each subplot of the figure plots a single estimate from 10 replicate trials (thin light-gray colored line) and shows their average result as a solid black line, while the red line gives the true theoretical density. It can be seen that when the coreset size is small (e.g., $k = 50$), uniform sampling is too random and often fails to cover both ends of the distribution, leading to significant deviations in the predicted curves; using only ℓ_2 leverage sampling, with a relatively small coreset size, there is still a relatively high risk of failure.; and introduction of the convex hull can effectively ensure fits the theoretical curves better at the minimum size. As k increases, the average predictions of all three methods gradually converge to the red true density - but at any scale, the ℓ_2 + convex hull scheme reproduces the main shape of the distribution earliest and most consistently, fully reflecting its ability to efficiently capture the distribution with small samples.

E.2 Real-world Experiments Results

E.2.1 Covertype Dataset

We demonstrate our method on the widely used UCI Covertype dataset (Blackard, 1998), originally intended for forest cover type classification. The full dataset comprises $n = 581\,012$ observations and 54 variables, among which 10 continuous variables represent various terrain features, such as elevation, slope, aspect, distances to hydrological features, and hillshade indices.

This dataset presents several practical challenges motivating the use of MCTM. Specifically, the continuous variables exhibit complex joint dependency structures characterized by multimodality, heavy skewness, and highly non-linear pairwise interactions. Standard Gaussian or parametric copula approaches typically fail to capture these features adequately, thus justifying the adoption of MCTM, which flexibly models the multivariate distribution through non-linear transformations.

However, fitting a full MCTM on large-scale datasets is computationally intensive and sometimes fails. For

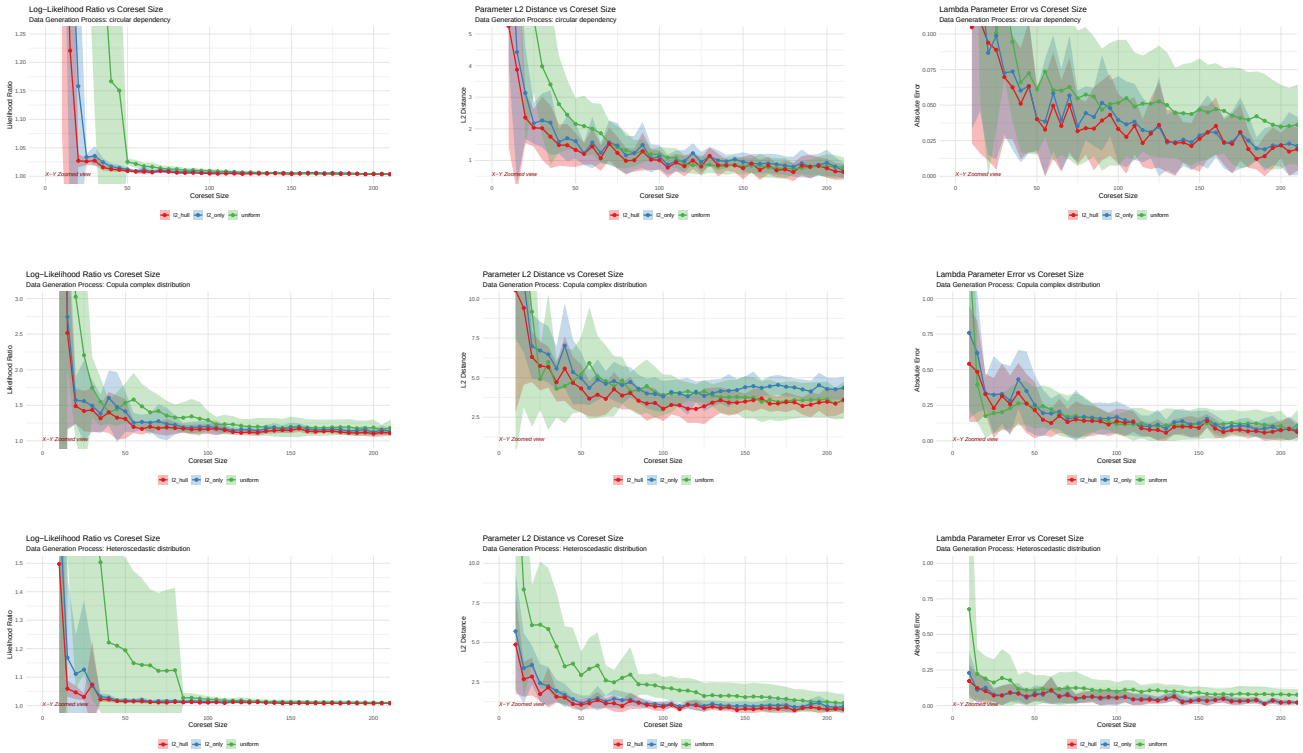


Figure 8: Convergence of the likelihood ratio, parameter error, and λ error as coreset size increases. First row: circular-dependency. Data points are distributed along circular trajectories, demonstrating a circular dependency structure between variables. Second row: copula complex. A copula construct with tail dependence and non-linear correlation is used. Third row: heteroscedastic distribution. The conditional variance of the data varies with the level of the independent variable, reflecting heteroscedasticity.

Figure 9: Computation time for 9 simulation distributions

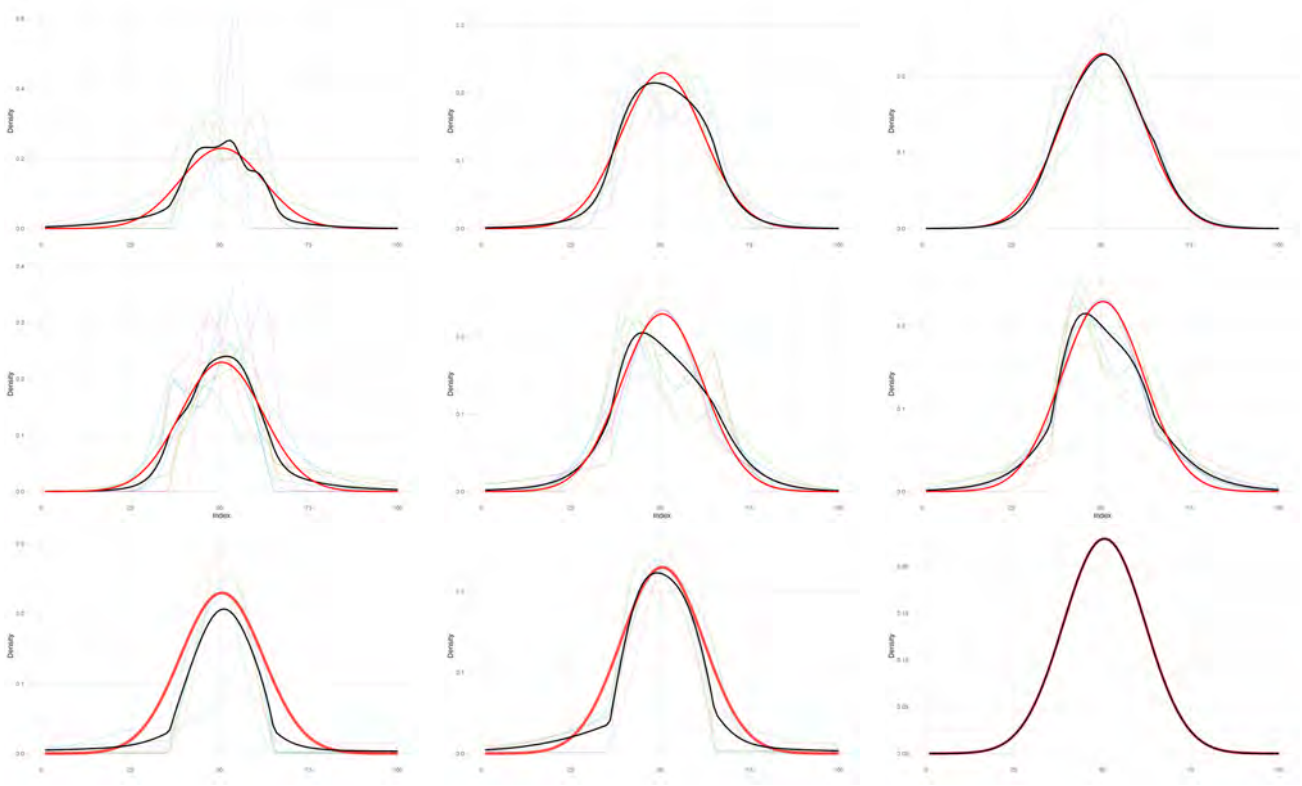


Figure 10: Predicted marginal density of x for the normal distribution of different coreset size, $k = 50, 100, 500$. First Row: uniform subsampling. Second row: Only ℓ_2 sampling. Third row: ℓ_2 with ε-kernel convex hull

example, in our case, on the Covertype dataset, when we attempted to fit the 10-dimensional MCTM model to the original $n = 581\,012$ size dataset, hardware limitations caused this fit to simply crash. Therefore, we selected a subsample of $n = 300\,000$ as the original model, and directly fitting an MCTM with only 10 continuous variables on $n = 300\,000$ observations already requires several hours of computation time. The computational challenge rapidly intensifies with higher dimensions or larger sample sizes, thus providing a natural and compelling scenario for demonstrating the practical benefits of our coreset approach.

We use the Covertype dataset to empirically assess whether coresets can efficiently approximate the full-data MCTM while substantially reducing computational costs, thereby enabling scalable probabilistic modeling of large multivariate datasets.

As can be seen from the Figure 13, our proposed ℓ_2-`hull` method significantly outperforms the uniform sampling (`uniform`) in all the four evaluation metrics. Specifically, ℓ_2-`hull` is closer to 1 in the log-likelihood ratio, which indicates a more accurate approximation of the original model; it maintains a smaller error in the ℓ_2 distance between the ϑ parameters and λ as well, which proves that it is able to better preserve the distributional structure of the original data in the parameter space whose dimension is roughly squared compared to the plain data dimension; and at the same time, its overall running time is comparable to that of uniform sampling, which does not introduce any additional significant overheads. It can be seen that for multivariate large-scale datasets, the ℓ_2-`hull` method, which combines ℓ_2 subsampling with convex hull techniques, outperforms simple uniform sampling in terms of both performance and efficiency.

E.2.2 Equity Return Dataset

In finance, returns on multiple stocks often have complex non-linear and time-varying dependence structures. To capture these dependencies, copula models are widely used in risk management and asset allocation. However, traditional copula mostly need to specify the marginal distributions and assume fixed dependence parameters, which makes it difficult to simultaneously model the dynamic changes of the marginals and conditional correlations. MCTM can more comprehensively reflect the characteristics of the joint distribution of financial asset

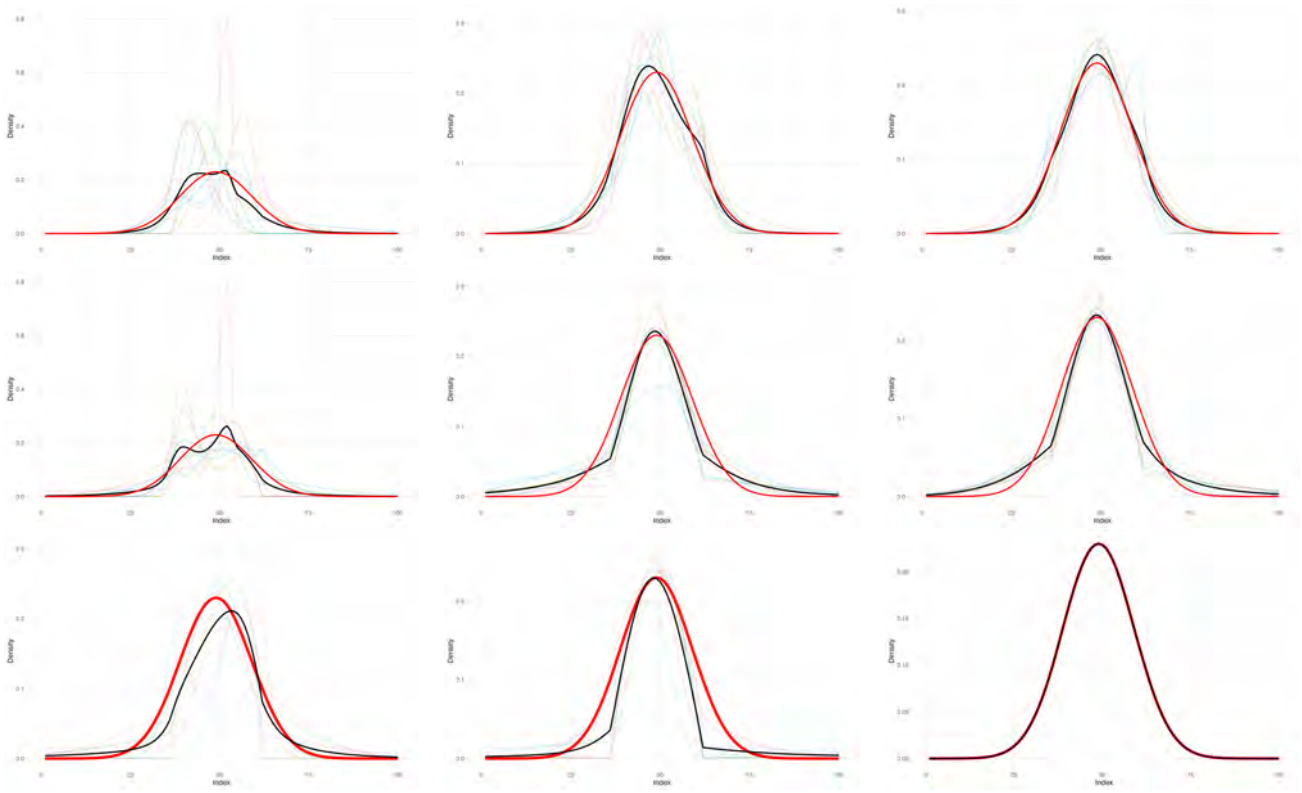
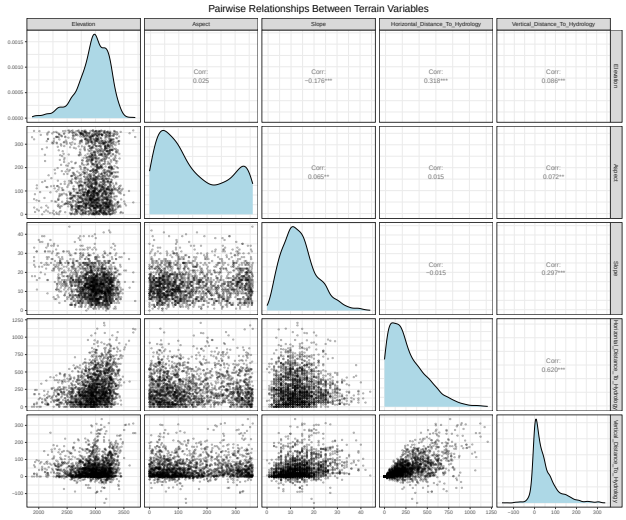


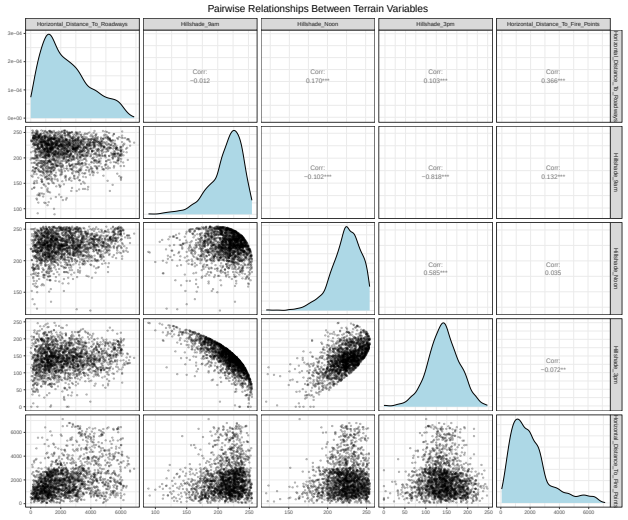
Figure 11: Predicted marginal density of y of different coreset size, $k = 50, 100, 500$. First Row: uniform subsampling. Second row: Only ℓ_2 sampling. Third row: ℓ_2 with ε-kernel convex hull

returns by jointly estimating the marginal distributions and the dependence structures through a flexible transformation function, and thus has a good prospect of being applied in the modeling of stock returns. Therefore, we select 10 and 20 representative stock returns data, see Table 7 and Table 8, respectively, construct their joint distributions based on the multivariate conditional transformation model MCTM, and combine the coreset method proposed in this paper to improve the computational efficiency of model fitting.

In Table 5 (10 stocks) and Table 6 (20 stocks) we can see that in most of the experimental scenarios, the ℓ_2-hull method achieves excellent performance, especially in the two metrics of ℓ_2 distance and log-likelihood ratio in the parameter space, which are comparable compared to the pure ℓ_2 sampling scheme. This may be due to the fact that in these scenarios, there are not many extreme points and the convex hull approximation does not add a significant advantage. In addition to this, the ℓ_2-hull is far superior to uniform subsampling in terms of parameter error and log-likelihood ratios in the vast majority of scenarios, and is no slouch in terms of estimation accuracy dependent on the structural parameter λ. In summary, the combination of sensitivity sampling and convex hull technique not only maintains the comparable effect with simple ℓ_2 sampling, but also outperforms uniform sampling when facing sparse extremes and complex multivariate structures, and achieves a good balance between high accuracy and high efficiency.



(a) Elevation, Aspect, Slope, Horizontal and Vertical Distance to Hydrology.



(b) Roadways, Hillshade at different times, and Fire Points.

Figure 12: Pairwise relationships between different sets of terrain variables from the Covertype dataset.

Table 5: Performance comparison on 10 stock return series for different coreset sizes (1985-2025)

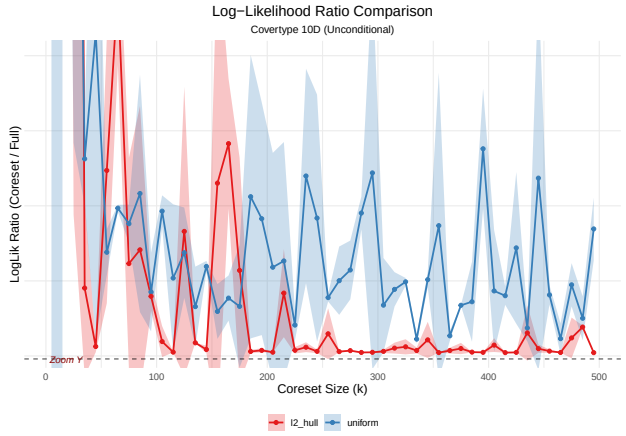
Coreset Size	Method	Param. ℓ_2 dist.	λ error	Log-likelihood ratio	Rel. Impr. L2 / λ / LL (%)	Total time (s)
$k = 50$	ℓ_2-hull	$\mathbf{45.518 \pm 1.643}$	2.992 ± 0.530	1.683 ± 0.234	12.0 / 0.0 / 57.5	8.21 ± 2.30
	ℓ_2-only	45.784 ± 0.554	2.599 ± 0.629	$\mathbf{1.351 \pm 0.082}$	11.5 / 0.0 / 78.2	8.45 ± 2.53
	uniform	51.745 ± 2.474	$\mathbf{1.515 \pm 0.303}$	2.610 ± 0.101	baseline	$\mathbf{7.98 \pm 1.88}$
$k = 100$	ℓ_2-hull	$\mathbf{39.888 \pm 0.643}$	1.613 ± 0.106	1.399 ± 0.147	22.1 / 0.0 / 89.3	9.93 ± 2.58
	ℓ_2-only	41.183 ± 1.503	1.404 ± 0.114	$\mathbf{1.153 \pm 0.044}$	19.6 / 0.0 / 96.1	9.80 ± 2.91
	uniform	51.195 ± 2.616	$\mathbf{0.913 \pm 0.140}$	4.926 ± 0.390	baseline	$\mathbf{8.94 \pm 2.10}$
$k = 200$	ℓ_2-hull	$\mathbf{33.722 \pm 1.242}$	1.050 ± 0.133	1.236 ± 0.145	29.6 / 0.0 / 80.9	11.37 ± 2.89
	ℓ_2-only	38.031 ± 0.282	1.147 ± 0.140	$\mathbf{1.085 \pm 0.004}$	20.6 / 0.0 / 93.1	11.40 ± 3.51
	uniform	47.881 ± 1.217	$\mathbf{0.621 \pm 0.108}$	2.242 ± 0.149	baseline	$\mathbf{10.13 \pm 2.69}$
$k = 300$	ℓ_2-hull	$\mathbf{29.814 \pm 1.363}$	0.905 ± 0.100	1.127 ± 0.081	33.6 / 0.0 / 92.1	13.95 ± 4.48
	ℓ_2-only	35.107 ± 0.916	0.851 ± 0.091	$\mathbf{1.070 \pm 0.007}$	21.8 / 0.0 / 93.5	$\mathbf{12.63 \pm 3.13}$
	uniform	44.915 ± 3.172	$\mathbf{0.488 \pm 0.050}$	2.079 ± 0.171	baseline	13.01 ± 3.91

Results are mean ±1 standard deviation over 5 valid trials. Bold indicates the best (lowest for error metrics, likelihood ratio closer for 1 for better) per coreset size and metric. Relative improvement (%) over the uniform baseline shown per metric (Param L2 / Lambda / LL)

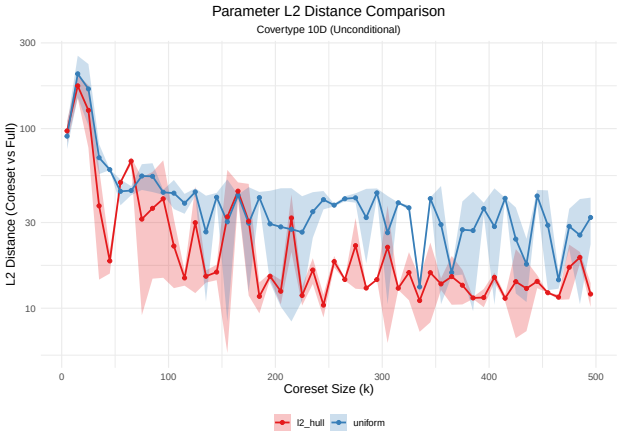
Table 6: Performance comparison on 20 stock return series for different coreset sizes (1985-2025)

Coreset Size	Method	Param. ℓ_2 dist.	λ error	Log-likelihood ratio	Rel. Impr. L2 / λ / LL (%)	Total time (s)
$k = 50$	ℓ_2-hull	54.254 ± 1.367	3.944 ± 0.563	5.208 ± 0.210	3.6 / 0.0 / 0.0	30.95 ± 5.06
	ℓ_2-only	$\mathbf{53.922 \pm 1.280}$	3.920 ± 0.344	30.303 ± 0.426	4.2 / 0.0 / 0.0	31.10 ± 6.00
	uniform	56.263 ± 1.042	$\mathbf{3.490 \pm 0.260}$	$\mathbf{4.504 \pm 0.200}$	baseline	$\mathbf{29.00 \pm 7.55}$
$k = 100$	ℓ_2-hull	$\mathbf{46.280 \pm 1.887}$	2.270 ± 0.138	$\mathbf{1.675 \pm 0.154}$	13.4 / 0.0 / 89.6	35.56 ± 7.94
	ℓ_2-only	47.385 ± 0.788	2.304 ± 0.370	1.718 ± 0.141	11.4 / 0.0 / 88.9	$\mathbf{34.95 \pm 6.53}$
	uniform	53.469 ± 2.072	$\mathbf{2.006 \pm 0.151}$	7.518 ± 0.305	baseline	35.22 ± 6.65
$k = 200$	ℓ_2-hull	40.225 ± 3.690	1.511 ± 0.138	1.574 ± 0.147	20.4 / 0.0 / 62.3	49.73 ± 7.74
	ℓ_2-only	$\mathbf{38.852 \pm 0.925}$	1.458 ± 0.079	$\mathbf{1.131 \pm 0.044}$	23.1 / 0.0 / 91.4	52.99 ± 12.33
	uniform	50.531 ± 1.513	$\mathbf{1.224 \pm 0.158}$	2.525 ± 0.067	baseline	$\mathbf{45.91 \pm 7.39}$
$k = 300$	ℓ_2-hull	$\mathbf{35.819 \pm 3.552}$	1.165 ± 0.099	1.400 ± 0.107	23.6 / 0.0 / 75.2	$\mathbf{60.99 \pm 6.86}$
	ℓ_2-only	35.036 ± 0.286	1.137 ± 0.132	$\mathbf{1.109 \pm 0.033}$	25.3 / 0.0 / 93.2	61.41 ± 10.14
	uniform	46.906 ± 3.669	$\mathbf{1.028 \pm 0.044}$	2.617 ± 0.195	baseline	63.89 ± 12.15

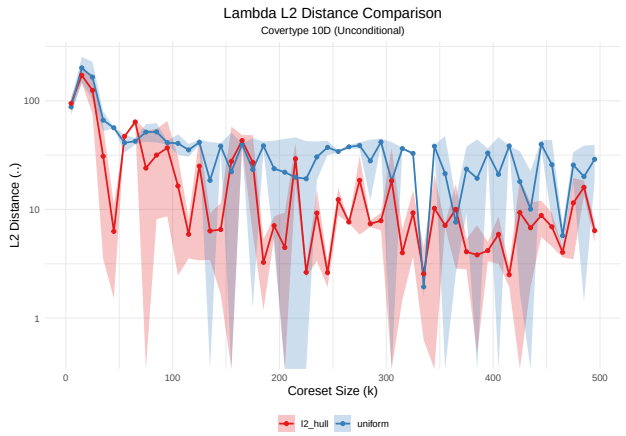
Results are mean ±1 standard deviation over 5 valid trials. Bold indicates the best (lowest for error metrics, likelihood ratio closer for 1 for better) per coreset size and metric. Relative improvement (%) over the uniform baseline shown per metric (Param L2 / Lambda / LL)



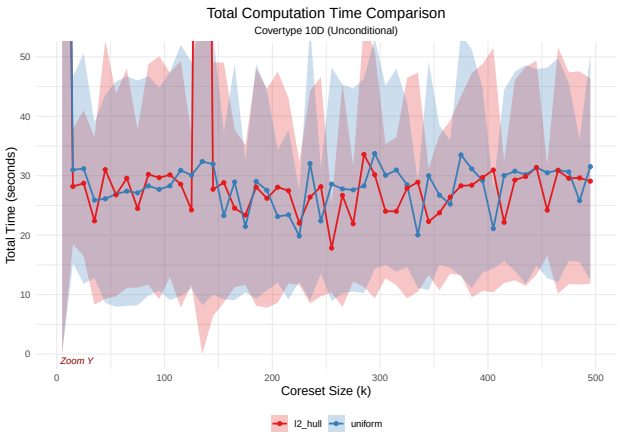
(a) Log-Likelihood Ratio Comparison



(b) Parameter ℓ_2 Distance Comparison



(c) Lambda ℓ_2 Distance Comparison



(d) Total Computation Time Comparison

Figure 13: Covertype dataset (10-dimensional, unconditional model) on ℓ_2-`hull` versus uniform sampling (`uniform`) compared with respect to four evaluation metrics: (a) log-likelihood ratio, (b) parameter-space ℓ_2 distances, (c) ℓ_2 distances dependent on parameter λ, (d) Total computation time.

Table 7: List of 10 Selected Stocks

Ticker	**Company Name**
JNJ	Johnson & Johnson
PG	Procter & Gamble Co.
KO	The Coca-Cola Company
XOM	Exxon Mobil Corporation
WMT	Walmart Inc.
IBM	International Business Machines Corporation
GE	General Electric Company
MMM	3M Company
MCD	McDonald's Corporation
PFE	Pfizer Inc.

Table 8: List of 20 Selected Stocks

Ticker	**Company Name**
JNJ	Johnson & Johnson
PG	Procter & Gamble Co.
KO	The Coca-Cola Company
XOM	Exxon Mobil Corporation
WMT	Walmart Inc.
IBM	International Business Machines Corporation
GE	General Electric Company
MMM	3M Company
MCD	McDonald's Corporation
PFE	Pfizer Inc.
AAPL	Apple Inc.
MSFT	Microsoft Corporation
INTC	Intel Corporation
CSCO	Cisco Systems Inc.
AMGN	Amgen Inc.
CMCSA	Comcast Corporation
COST	Costco Wholesale Corporation
GILD	Gilead Sciences Inc.
SBUX	Starbucks Corporation
TOT	TotalEnergies SE