Multivariate Density Estimation
Comparing Transformation Random Forests, Triangular Transport Maps, and Copulas

---

Master's Thesis in Biostatistics (STA495)

by

Léon Kia Faro
13-795-026

supervised by

Prof. Dr. Torsten Hothorn

Zurich, September 2025

Abstract

This thesis evaluates three approaches to multivariate density estimation for tabular data within a single, consistent pipeline: separable triangular transport maps (TTM-Sep), Transformation Random Forests (TRTF), and copulas. All methods use standardized inputs and a common evaluation protocol so that likelihoods, diagnostics, and compute are directly comparable. In the configuration studied, TRTF and TTM-Sep yield the same triangular-likelihood form, which enables like-for-like evaluation.

On Half-Moon ($n = 250$), mean joint negative log-likelihoods (NLL; lower is better) were 1.71 (TRTF), 1.93 (TTM-Sep), and 1.54 (copula). On a four-dimensional autoregressive generator they were 4.53, 5.66, and 5.45, respectively; permutation averages confirm order sensitivity for triangular maps. On MiniBooNE ($K = 43$; sum test log-likelihood), TRTF reached -30.01 under the standard preprocessing and training budget used here; published flow models report values around -12 to -16 under their settings. These numbers are not strictly comparable but indicate the relative accuracy of this configuration.

Overall, TRTF tends to lead within the separable family at low dimension, while higher-dimensional datasets expose the limits of separable structure. We report robustness checks (ordering), calibration diagnostics, and the numerical safeguards used, and we outline directions toward richer parameterizations within the same evaluation frame.

Contents

Chapter 1

Introduction

Multivariate density estimation is a cornerstone of modern statistics and machine learning. It supplies the machinery for likelihood-based inference, simulation of generative mechanisms, and uncertainty quantification in tabular domains. Practitioners, however, routinely face a delicate trade-off between three goals: expressive power, computational feasibility, and interpretability. Three complementary families of estimators have emerged around this dilemma:

1. **Normalizing flows** build expressive change-of-variables maps and have become central in deep generative modeling [Rezende and Mohamed, 2015, Papamakarios et al., 2017, Dinh et al., 2017, Durkan et al., 2019, Kingma and Dhariwal, 2018, Papamakarios et al., 2021].

2. **Transformation models and forests** model conditional distributions through parametric transformation structures coupled with ensemble methods [Hothorn and Zeileis, 2017, Hothorn et al., 2018, Hothorn and Zeileis, 2021].

3. **Copulas** separate marginals from dependence, yielding interpretable multivariate association models [Sklar, 1959, Nelsen, 2006, Joe, 2014, Nagler, 2017].

This thesis situates all three families inside a common **transport framework** that makes them directly comparable [Rosenblatt, 1952, Knothe, 1957, Bogachev et al., 2005, Ramgraber et al., 2025]. The framework operates in standardized coordinates u computed solely from training-split statistics to avoid leakage and keep objectives aligned across models. Within this evaluation framework each estimator couples u to a reference distribution—Gaussian for triangular maps and TRTF, $\mathrm{Unif}([0,1]^K)$ for copulas—and recovers densities via exact change of variables. Section 1.2 previews the core pieces and Chapter 2 develops the mathematics in detail.

Separable triangular transport maps play a central role throughout the thesis. Their lower-triangular Jacobian delivers a log-determinant in $\mathcal{O}(K)$ operations, back-substitution yields an explicit inverse for sampling, and the triangular structure produces exact conditional densities in a fixed order. Fully triangular maps can require $\mathcal{O}(K^2)$ work per evaluation, whereas separable variants reduce both forward and inverse passes to $\mathcal{O}(K)$, making them attractive for systematic comparison while preserving interpretable marginals and dependence mechanisms.

Notation. We write φ and Φ for the density and cumulative distribution function of a univariate standard normal random variable, and η for the K-variate standard normal density. All models are evaluated in standardized coordinates u and reported on the original scale x through a diagonal affine correction so that log densities remain comparable. Unless stated otherwise, log values are expressed in natural units (nats).

1.1 Thesis and Problem Statement

The goal of this thesis is to examine multivariate density estimation for tabular data within a unified transport-based evaluation protocol. We focus on three estimators: separable triangular transport maps (TTM-Sep), Transformation Random Forests (TRTF), and nonparametric copula models.

Let $u = D^{-1}(x - \mu)$ with $D = \mathrm{diag}(\sigma_1, \ldots, \sigma_K)$ determined from the training split. If $T : \mathbb{R}^K \to \mathbb{R}^K$ is lower triangular, $z = T(u)$, and η denotes the chosen reference density, then the change-of-variables formula gives

$$\log p_X(x) = \log \eta\big(T(u)\big) + \sum_{k=1}^{K} \log \frac{\partial T_k}{\partial u_k}(u_{1:k}) - \sum_{k=1}^{K} \log \sigma_k, \tag{1.1}$$

so reporting on the x-scale differs from reporting on the u-scale by the constant $-\sum_k \log \sigma_k$ per data point. The Rosenblatt and Knothe rearrangements guarantee the existence of such triangular transports for any coordinate ordering [Rosenblatt, 1952, Knothe, 1957].

The copula baselines considered here are nonparametric kernel copulas [Nagler, 2017]. We fit smooth univariate marginals with `kde1d`, form pseudo-observations using mid-rank empirical CDFs during training to avoid double smoothing, and evaluate joint densities through

$$p_X(x) = c\big(F_1(x_1), \ldots, F_K(x_K)\big) \prod_{k=1}^{K} f_k(x_k),$$

where c is the copula density and $F_k = \int f_k$ the smoothed marginal CDFs. No Gaussian copula enters the study; the emphasis remains on flexible, interpretable diagnostics in low dimensions.

Separable triangular maps decompose each coordinate into a context-dependent shift plus a one-dimensional monotone transform. TRTF occupies an intermediate position: under strictly increasing conditional CDFs and the standard forest aggregation after monotone smoothing it reproduces the same separable triangular likelihood via the probability integral transform. The shared protocol described above keeps preprocessing, likelihood evaluation, calibration diagnostics, and compute measurements aligned across all estimators.

1.2 The Transport Frame on One Page

The key ingredients of the transport framework are:

- **Standardization.** Coordinates $u = D^{-1}(x - \mu)$ use training-split statistics only; all logs are in nats.
- **Change of variables.** Densities follow from $z = T(u)$ through Equation (1.1).
- **Triangular log determinant.** Lower-triangular Jacobians imply $\log\det J_T(u) = \sum_k \log \partial_{u_k} T_k(u_{1:k})$, giving $\mathcal{O}(K)$ cost.
- **Affine correction.** Reporting on x applies the constant correction $-\sum_k \log \sigma_k$ per data point.
- **Computational profile.** Fully triangular maps can incur $\mathcal{O}(K^2)$ evaluation and inversion, whereas separable maps reduce both passes to $\mathcal{O}(K)$ while keeping the log determinant linear in K.

Separable triangular transports implement each coordinate as the sum of a context shift and a monotone one-dimensional basis, maintaining determinant stability and consistent conditional behaviour. TRTF achieves the same likelihood under the conditions noted above, while nonparametric copulas operate on the unit hypercube with explicit marginals. Standardization $u = T_{\text{std}}(x)$ feeds either the triangular transport branch (TTM variants, TRTF) or the copula branch, and both return log densities, conditional diagnostics, samples, calibration metrics, and compute summaries.

1.3 Contributions and Research Questions

The thesis contributes along three axes that move from theoretical unification to empirical benchmarking and practical interpretation:

1. **Unified likelihood framework.** We formalise a shared transport perspective that places separable triangular maps, TRTF, and nonparametric copulas inside one mathematical scheme. The equivalence between TRTF and separable triangular transports holds under explicit regularity assumptions—strictly monotone conditional CDFs and the forest aggregation after smoothing.

   > **Proposition (informal).** Under axis-parallel partitions, strictly monotone conditional CDFs, and the forest aggregation after smoothing, the TRTF likelihood coincides with that of a separable triangular transport in standardized coordinates.

2. **Benchmarking under a common protocol.** We design a standardized evaluation pipeline covering synthetic generators and real tabular benchmarks. TTM-Sep, its marginal variant, and TRTF are evaluated systematically; nonparametric kernel copulas act as diagnostic comparators when $K \leq 3$. The pipeline produces three categories of evidence: (i) average test log likelihoods, (ii) conditional diagnostics based on probability integral

Table 1.1: Model abbreviations used throughout the thesis.

Label	Meaning
TTM-Marg	Marginal triangular transport (per-dimension; no context)
TTM-Sep	Separable triangular transport (context shift + monotone 1D transform)
TRTF	Transformation Random Forests (axis-parallel splits)
True-Marg	Oracle marginal density
True-Joint	Oracle conditional joint density
Copula	Nonparametric kernel copula baseline (smoothed marginals; no Gaussian copula)

transforms, and (iii) compute measurements for training effort and per-sample evaluation time. Together they quantify the trade-offs among accuracy, calibration, and efficiency.

3. **Operational insights.** The theoretical and empirical findings yield guidance for practitioners. We analyse sensitivity to ordering, the practical limits of separability, and when nonparametric copulas remain informative as diagnostic references. Chapter 4 synthesizes these insights and outlines extensions toward richer parameterisations.

Two research questions organise the empirical study:

- **Synthetic data.** How do TRTF, TTM-Sep, and nonparametric copulas compare on controlled generators? Evidence comprises likelihood comparisons, conditional calibration metrics, and timing measurements.
- **Benchmark datasets.** How close do TRTF results come to published normalizing flow baselines under identical preprocessing? Side-by-side likelihood comparisons, interpreted through the constraints of separability and compute profiles, answer this question.

Addressing these questions bridges rigorous transport-based theory with empirical validation, clarifying when simple triangular models suffice and when more expressive families become necessary.

Chapter 2

Methodological Background

2.1 Transport Framework and Notation

This section establishes the coordinate system, notation, and algebraic identities used throughout the thesis. The schematic in Appendix A remains valid; here we focus only on the formulas that are needed later. We set up the standardized pullback likelihood, the triangularity assumption, and the Jacobian factorization.

We write $u_{1:k} = (u_1, \ldots, u_k)$ and use ∂_{u_k} for partial derivatives with respect to u_k.

We observe data on the original scale $x \in \mathbb{R}^K$. Training-split statistics define a fixed standardization map

$$u \;=\; T_{\mathrm{std}}(x) \;=\; (x - \mu) \oslash \sigma, \qquad \sigma_k > 0, \tag{2.1}$$

where μ and σ are the empirical mean and standard deviation estimated on the training split, and $\oslash$ denotes elementwise division. Each feature is shifted and rescaled once using training data only. All derivatives and Jacobians are then taken in u-space, which avoids leakage and keeps results comparable across estimators.

We model π_U as the pullback of a simple reference η through a triangular map $S : u \mapsto z$. The reference is the K-variate standard normal $\eta(z)$. Applying the change-of-variables formula gives

$$\pi_U(u) \;=\; \eta(S(u)) \, \left|\det \nabla_u S(u)\right|. \tag{2.2}$$

To report log densities on the original scale, we apply only the diagonal affine correction implied by standardization:

$$\log \pi_X(x) \;=\; \log \pi_U(T_{\mathrm{std}}(x)) - \sum_{k=1}^{K} \log \sigma_k. \tag{2.3}$$

The transport is assumed to be lower triangular and strictly increasing in each coordinate,

$$S(u) \;=\; \big(S_1(u_1), S_2(u_{1:2}), \ldots, S_K(u_{1:K})\big), \qquad \partial_{u_k} S_k(u_{1:k}) > 0, \tag{2.4}$$

Table 2.1: Notation for the transport framework used in Chapters 2 and 3. All derivatives and Jacobians are with respect to u; log densities on x apply the affine correction in Equation (2.3).

Symbol	Meaning
$x \in \mathbb{R}^K$	Original features on the data scale
T_{std}	Standardization map using training (μ, σ)
$u = T_{\text{std}}(x)$	Standardized evaluation coordinates
$z \in \mathbb{R}^K$	Reference coordinates after transport
$S : u \mapsto z$	Monotone lower-triangular transport map
$\nabla_u S(u)$	Jacobian of S with respect to u
$\eta(z)$	K-variate standard normal density
$\varphi(t)$, $\Phi(t)$	Univariate standard normal density and CDF
π_U, π_X	Densities on u- and x-space
μ, σ	Training mean vector and positive scales
K	Dimension of the feature vector

so the Jacobian $\nabla_u S(u)$ is lower triangular. Its determinant factorizes as

$$\log \left| \det \nabla_u S(u) \right| \;=\; \sum_{k=1}^{K} \log \partial_{u_k} S_k(u_{1:k}). \tag{2.5}$$

This factorization makes the *log-determinant* cost $\mathcal{O}(K)$ once the diagonal terms are available. Evaluating all $S_k(u_{1:k})$ can be $\mathcal{O}(K^2)$ for a general triangular map, but reduces to $\mathcal{O}(K)$ in the separable parameterization of Section 2.2.

Strict monotonicity on each coordinate ($\partial_{u_k} S_k > 0$) together with triangularity implies that S is a bijection with a triangular inverse, computable by back-substitution.

Taken together, the triangular and strictly monotone structure turns joint density evaluation and sampling into sequential one-dimensional stages, and the standardized coordinates allow us to report on the x-scale with only the diagonal correction from Equation (2.3).

2.2 Separable Triangular Maps and Transformation Random Forests

We now move from the general triangular setup to the first concrete parameterization: the **separable triangular map**. We then show how **Transformation Random Forests (TRTF)** fit into the same framework. The aim is a shared likelihood that we can train, compare, and diagnose across estimators without changing conventions.

2.2.1 Separable triangular component

A separable component has the form

$$S_k(u_{1:k}) \;=\; g_k(u_{1:k-1}) + h_k(u_k), \qquad h_k'(u_k) > 0, \tag{2.6}$$

where earlier coordinates shift location through g_k, while h_k shapes the marginal and ensures monotonicity. To fix identifiability, we enforce a convention such as centering g_k on the training split, $\mathbb{E}[g_k(U_{1:k-1})] = 0$.

With the standard normal reference, the pullback identity becomes

$$\log \pi_U(u) \;=\; \sum_{k=1}^{K} \Big[\log \varphi\big(S_k(u_{1:k})\big) + \log h_k'(u_k) \Big]. \tag{2.7}$$

Up to a constant, the per-sample negative log-likelihood is

$$\mathcal{L}(u) \;=\; \sum_{k=1}^{K} \Big[\tfrac{1}{2}\, S_k(u_{1:k})^2 - \log h_k'(u_k) \Big]. \tag{2.8}$$

For a dataset $\{u^{(i)}\}_{i=1}^{n}$, the empirical objective reads

$$\mathcal{L}_n = \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K} \Big[\tfrac{1}{2}\, S_k(u_{1:k}^{(i)})^2 - \log h_k'(u_k^{(i)}) \Big] + \text{const.}$$

These formulas evaluate in $\mathcal{O}(K)$ time per sample, since the Jacobian term depends only on u_k. Sampling inverts S by back-substitution, and log densities on x-space apply the affine correction in Equation (2.3). The speed comes with a trade-off: separability fixes the conditional shape along u_k once g_k shifts the location, so variance, skewness, and modality no longer adapt to the context $u_{1:k-1}$.

2.2.2 Transformation Random Forests in the framework

Transformation Random Forests (TRTF) fit conditional CDFs and insert them into the triangular pipeline via the probability integral transform. Let $\widehat{F}_k(\cdot \mid u_{1:k-1})$ denote the smoothed conditional CDF. Define

$$S_k(u_{1:k}) \;=\; \Phi^{-1}\Big(\widehat{F}_k(u_k \mid u_{1:k-1}) \Big). \tag{2.9}$$

If the forest is parameterized additively as

$$\widehat{F}_k(u_k \mid u_{1:k-1}) \;=\; \Phi\big(h_k(u_k) + g_k(u_{1:k-1})\big), \tag{2.10}$$

then we recover the separable structure,

$$S_k(u_{1:k}) \;=\; h_k(u_k) + g_k(u_{1:k-1}), \qquad h_k'(u_k) > 0. \tag{2.11}$$

Derivative check.

$$\partial_{u_k} S_k = \frac{d}{du_k} \Phi^{-1}(\Phi(h_k(u_k) + g_k)) = h_k'(u_k) > 0,$$

so the Jacobian diagonal is indeed $h_k'(u_k)$ in the additive TRTF case.

Numerical note. In practice, forests may produce $\widehat{F}_k$ values outside $(0,1)$ by tiny amounts or with imperfect monotonicity. We therefore clip to $(\varepsilon, 1-\varepsilon)$ with $\varepsilon = 10^{-6}$ and, if needed, apply isotonic post-processing to enforce strict increase. This keeps $S_k = \Phi^{-1}(\widehat{F}_k)$ well defined.

With this forest aggregation and strictly monotone conditionals, the resulting TRTF map coincides with the separable parameterization, so it delivers the same likelihood and back-substitution sampler.

2.3 Copula Baselines

We now turn to copulas, used here as diagnostic baselines rather than high-dimensional competitors. The focus is on a nonparametric bivariate copula when $K = 2$, with an independence fallback otherwise. Standardization remains global (based on the full training split), whereas marginal fits for copulas are per-class. This ensures comparability on the x-scale when reporting.

2.3.1 Estimation and reporting

For each class y, we proceed as follows:

- **Marginals:** Fit univariate kernel densities (`kde1d`) to obtain $\widehat{f}_k(\cdot \mid y)$ and CDFs $\widehat{F}_k(\cdot \mid y)$.
- **Pseudo-observations:** Transform samples to $(0,1)$ using mid-ranks,

$$U_{ik}^{(y)} \;=\; \frac{\operatorname{rank}(X_{ik}^{(y)})}{n_y + 1}. \tag{2.12}$$

  Mid-ranks handle ties; if the data contain many, a tiny jitter is added before ranking.

- **Dependence:** Fit a bivariate kernel copula density $c_y(u_1, u_2)$ using `kdecopula::kdecop` with local quadratic smoothing, which uses a transformation approach to avoid boundary bias [Nagler, 2017].

For prediction at $x = (x_1, x_2)$:

1. Evaluate $\widehat{f}_k(x_k \mid y)$ and $\widehat{F}_k(x_k \mid y)$.
2. Evaluate $c_y(\widehat{F}_1(x_1 \mid y), \widehat{F}_2(x_2 \mid y))$.
3. Combine them through

$$\log \widehat{\pi}_X(x \mid Y = y) \;=\; \log \widehat{f}_1(x_1 \mid y) + \log \widehat{f}_2(x_2 \mid y) + \log c_y\Big(\widehat{F}_1(x_1 \mid y), \widehat{F}_2(x_2 \mid y)\Big). \tag{2.13}$$

When labels are unknown, we mix across classes using priors $\pi(y)$ and a stable log-sum-exp.

2.3.2 Fallback and scope

If $K \neq 2$, labels are absent, or packages unavailable, we fall back to an independence model:

$$\log \widehat{\pi}_X^{\text{ind}}(x) \;=\; \sum_{k=1}^{K} \log \widehat{f}_k(x_k). \tag{2.14}$$

Hence the NP-Copula is restricted to $K = 2$ and is used only as a diagnostic baseline.

2.3.3 Consistency with evaluation frame

Copulas operate directly on the original scale x and therefore bypass the affine correction of Section 2.1. Still, the logic is parallel to the triangular branch: start from a simple reference (independent uniforms), add a dependence correction, and return a single joint log density. Using mid-ranks for fitting and $\widehat{F}_k$ for reporting is consistent with Sklar's theorem [Sklar, 1959, Nelsen, 2006, Joe, 2014].

2.3.4 Terminology and references

We use *NP-Copula* for the nonparametric kernel estimator and *Independence* for the fallback KDE product. References include Sklar's theorem and standard copula texts [Sklar, 1959, Nelsen, 2006, Joe, 2014], kernel copula methodology [Nagler, 2017], and the transformation-model literature for context [Hothorn and Zeileis, 2021]. In practice, the NP-copula baseline for $K = 2$ combines per-class marginals with a dependence correction, providing a transparent diagnostic comparator to the triangular transport models.

Chapter 3

Data Analysis and Validation

Motivation. We factor the joint density into simple conditional pieces using a lower-triangular parameterization. This yields exact likelihoods, transparent conditionals we can inspect, and $\mathcal{O}(K)$ per-sample cost. The shared factorization lets us run ordering experiments without changing the model class.

Model abbreviations. We use the short labels TTM-Sep (separable triangular transport map), TTM-X (separable map with low-rank cross-terms), TRTF (Transformation Random Forests; spelled out on first mention), RealNVP, MAF, NSF, True-Marg/True-Joint, and Copula consistently across tables and figures.

Units and signage. Unless stated otherwise, all log-likelihood (LL) and negative log-likelihood (NLL) values are reported in nats. Plots and tables mark LL (↑ better) and NLL (↓ better) explicitly.

Notation recap.

- Elementwise standardization with train-split statistics:
$$u_k = \frac{x_k - \mu_k}{\sigma_k}, \quad k = 1, \dots, K \qquad \text{equivalently } u = \operatorname{diag}(\sigma)^{-1}(x - \mu).$$
We compute derivatives and Jacobians in u-coordinates and transform back to x only when reporting densities on the original scale via Equation (2.3).
- Prefix subvector shorthand: $u_{1:k} := (u_1, \dots, u_k)$.
- Derivatives use $h'_k(\cdot)$ for univariate derivatives; mixed partials follow the convention from Chapter 2.

Copula conventions.

- All copula fits estimate marginals; reported values are joint densities on the original x-scale.
- Half-Moon ($K = 2$) uses class-conditional copulas for blue/red labels mixed by empirical class priors when we report the unlabeled density. No other model uses labels in this chapter; detailed results appear in Section 3.4.

3.1 Datasets and Preprocessing

This section fixes data sources, synthetic generators, and preprocessing so that objectives, diagnostics, and reported log-densities are comparable across models. All estimators operate in standardized coordinates, evaluate Jacobians in that space, and report log densities on the original scale using the affine correction from Equation (2.3). Throughout, logistic gates are written logistic$(\cdot)$.

3.1.1 Standardization and Splits

Features are standardized with training-split statistics only ($\mu, \sigma \in \mathbb{R}^K, \sigma_k > 0$) as in Equation (2.1). The triangular pullback, determinant factorization, and affine correction from Equations (2.2)–(2.3) stay in u-coordinates; we transform back to x once per evaluation when reporting densities. We keep fixed train/test splits. Synthetic calibration uses an 80/20 train/test partition. Real-data splits follow the published convention for each benchmark, and compute profiles are summarized in Appendix A.1.6.

For synthetic runs we hold seeds $\{11, 13, 17, 19, 23\}$ fixed across models and report means with ± 2 standard errors when averaging replicates. Ordering experiments reuse the lower-triangular factorization and vary only the coordinate order; the permutation sweeps in the four-dimensional study currently pin `SEED` $= 42$ unless overridden so that every ordering shares the same draws. Unless noted otherwise, our canonical ordering remains $(1, 2, 3, 4)$ for the four-dimensional generator; Section 3.4 reports the permutation studies.

3.1.2 Half-Moon ($K = 2$)

Half-Moon is a curved, bimodal joint designed to surface dependence beyond linear correlation. We evaluate joint log-density contours per model and compare results to the true joint. Figure 3.1 and Table 3.1 later in this chapter instantiate the diagnostics. For the copula baseline we fit blue/red class-conditional copulas and mix them by empirical class priors when reporting the unlabeled density. No other estimator uses labels.

3.1.3 Four-dimensional Autoregressive Generator ($K = 4$)

The synthetic autoregressive generator matches the lower-triangular factorization while stressing skewness, heavy tails, and context-dependent shape:

- $X_1 \sim \mathcal{N}(0, 1)$.
- $X_2 \sim \text{Exponential}(\lambda_0)$ with $\lambda_0 = 1$.
- $X_3 \mid X_{1:2}$ is a mixture of two beta laws on $(0, 1)$, e.g. $\text{Beta}(2.5, 5.0)$ versus $\text{Beta}(5.0, 2.5)$, with gate

$$w_3(X_{1:2}) = \text{logistic}\left(\gamma_0 + \gamma_1 X_1 + \gamma_2 (X_2 - 1)\right), \qquad \gamma = (\gamma_0, \gamma_1, \gamma_2).$$

- $X_4 \mid X_{1:3}$ is a mixture of two gamma laws on $(0, \infty)$ with distinct shape/scale pairs, gated by

$$w_4(X_{1:3}) = \text{logistic}\left(\delta_0 + \delta_1 X_1 + \delta_3(X_3 - 0.5)\right), \qquad \delta = (\delta_0, \delta_1, \delta_3).$$

Mixture weights are normalized through the usual softmax over component logits so that Jacobian terms stay well behaved under shared standardization. This design keeps X_4 conditionally heteroskedastic with positive support and aligns with the triangular evaluation cost $\mathcal{O}(K)$.

Why these generators? They match the triangular modeling assumptions yet stress estimators with skewness, heavy tails, and context-dependent variation. The true joint density and conditionals are known, which enables direct comparisons without confounding from Monte Carlo error.

Reproducibility note. All random seeds, splits, and generator parameters are fixed in the code release. Negative NLL values appear for bounded or highly concentrated marginals (e.g., beta or gamma mass near zero); this behavior is expected and does not signal an error.

Next: Section 3.2 details the estimators and training settings that make these likelihoods and diagnostics comparable across methods.

3.2 Models and Implementation

> **Scope.** This section specifies the estimators and the implementation details that keep our likelihoods, diagnostics, and plots comparable. We reuse the preprocessing and transport notation in Table 2.1, draw background from Chapter 2, follow the data handling in Section 3.1, and lean on the auxiliary pseudo-code in Appendix A.

3.2.1 Separable Triangular Transport Maps (TTM-Sep)

We use lower-triangular, separable transport maps

$$S_k(u_{1:k}) = g_k(u_{1:k-1}) + h_k(u_k), \qquad \partial_{u_k} S_k(u_{1:k}) = h_k'(u_k) > 0, \tag{3.1}$$

so the Jacobian contribution at stage k depends only on u_k. This structure yields $\mathcal{O}(K)$ per-sample cost and exact inversion by back-substitution.

Objective. We minimize the Gaussian pullback induced by the shared reference (Equations (2.2)–(2.5)):

$$\mathcal{L}(u) = \sum_{k=1}^{K} \left[\tfrac{1}{2} S_k(u_{1:k})^2 - \log h_k'(u_k)\right]. \tag{3.2}$$

We solve this objective with constrained optimization and enforce monotonicity by construction.

Parameterization.

- h_k: one-dimensional monotone bases (integrated radial basis functions, splines, and linear tails) with derivative nonnegativity so that $h_k' > 0$; all coefficients are learned jointly under the positivity constraint enforced by the optimizer.
- g_k: low-degree polynomial features of $u_{1:k-1}$ with light ridge regularization to keep $\nabla_u S(u)$ stable.

Ordering. Feature ordering matters for triangular models. We store the learned order and reuse it at prediction time. For permutation studies (Section 3.4) we apply the same bookkeeping so that permutations are reproducible.

3.2.2 Cross-term Variant (TTM-X)

We reference a low-rank interaction variant only to delimit the class used in the synthetic analyses:

$$S_k(u_{1:k}) = g_k(u_{1:k-1}) + h_k(u_k) + \sum_{j<k} \alpha_{kj}\, q_j(u_j)\, r_k(u_k), \tag{3.3}$$

where q_j and r_k are monotone features. Constraints ensure $\partial_{u_k} S_k > 0$ despite the cross-term. We do not promote TTM-X as a baseline; the definition clarifies the naming and scope used in the synthetic results.

3.2.3 Transformation Random Forests (TRTF)

We use Transformation Random Forests (TRTF) as an alternative route to a triangular component from conditional CDFs [Hothorn and Zeileis, 2017, Hothorn et al., 2018]. Let $\widehat{F}_k(\cdot \mid u_{1:k-1})$ denote the forest CDF. We define

$$S_k(u_{1:k}) = \Phi^{-1}\big(\widehat{F}_k(u_k \mid u_{1:k-1})\big), \tag{3.4}$$

which maps u_k into the standard normal reference conditional on $u_{1:k-1}$. By the chain rule,

$$\varphi\big(S_k(u_{1:k})\big)\, \partial_{u_k} S_k(u_{1:k}) = \widehat{f}_k(u_k \mid u_{1:k-1}),$$

so the TRTF-induced Jacobian matches the forest's conditional density. The triangular structure mirrors TTM-Sep; the difference is operational (forest training and aggregation) rather than conceptual.

3.2.4 Copula Baseline with Estimated Marginals

We keep copulas as dependence baselines with explicit scope. Marginals are estimated on the training split, and we report the joint density on the original x-scale. Concretely, we fit $\widehat{F}_k$ and $\widehat{f}_k$ for each coordinate, transform to pseudo-observations $U_k = \widehat{F}_k(X_k)$, and, for Gaussian copulas, set $Z_k = \Phi^{-1}(U_k)$ to model dependence in a latent normal space. We evaluate the copula density $c(U)$ and multiply by $\prod_{k=1}^{K} \widehat{f}_k(x_k)$ to obtain the reported joint density.

Label use (Half-Moon only). For $K = 2$ in the Half-Moon study we fit class-conditional copulas for the blue/red classes and mix them by empirical class priors when reporting the unlabeled joint. No other model uses labels. Details and diagnostics appear in Section 3.4.

3.2.5 Shared Inversion and Evaluation

We adopt a single inversion and evaluation convention across estimators using Equations (2.2)–(2.5). The alignment is necessary for conditional decomposition checks in Section 3.4 and for the compute summaries in Section 3.5.

Reproducibility. We fix random seeds, cache splits, and expose the same object layouts for transport maps, TRTF, and copulas. Appendix A provides pseudo-code for TRTF fitting and prediction; we reuse the same logging, timing hooks, and file formats across models.

3.3 Evaluation Metrics and Protocol

> **Goal.** Define the metrics and the evaluation routine applied across all estimators. We reuse the notation and preprocessing conventions from Table 2.1, with background in Chapter 2 and dataset preparation in Section 3.1. All log-likelihood (LL) and negative log-likelihood (NLL) values are reported in nats; figures and tables state LL (↑ better) and NLL (↓ better) explicitly.

3.3.1 Joint and Conditional Decomposition

Triangular models evaluate densities in standardized coordinates and then report on the original scale. With $u = T_{\text{std}}(x)$ determined by the training-split (μ, σ), any estimator with lower-triangular S satisfies

$$\log \hat{\pi}_U(u) = \sum_{k=1}^{K} \Big[\log \varphi\big(S_k(u_{1:k})\big) + \log \partial_{u_k} S_k(u_{1:k}) \Big], \tag{3.5}$$

so the determinant factorization from Equations (2.2)–(2.5) applies componentwise. We transform back to x once per evaluation via the affine correction in Equation (2.3).

We localize error with per-dimension conditional NLLs. For coordinate k,

$$\text{NLL}_k = -\frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \log \hat{\pi}\big(x_{ik} \mid x_{i,1:k-1}\big), \tag{3.6}$$

and the joint NLL satisfies $\text{NLL}_{\text{joint}} = \sum_{k=1}^{K} \text{NLL}_k$ by the triangular factorization. Negative per-dimension NLLs can occur on bounded or highly concentrated supports (e.g., beta or gamma margins); we revisit this behaviour in Section 3.4.

3.3.2 Compute Metrics

We report practical costs alongside fit:

- **Train time per epoch** and total wall-clock to the chosen early-stopping criterion when applicable.
- **Evaluation throughput** (test points per second) together with per-sample latency.
- **Model size,** recorded as effective parameter counts when the backend exposes them.

Compute summaries accompany the real-data tables in Section 3.5; extended profiles appear in Appendix A when needed.

3.3.3 Protocol (Fixed Across Methods)

1. **Preprocessing.** Standardize features with training-split (μ, σ), evaluate derivatives and Jacobians in u, and convert to x with Equation (2.3) once per evaluation.
2. **Evaluation in standardized space.** Compute LL, NLL, and conditional decompositions with Equation (3.5); map results to the original scale for reporting.
3. **Aggregation and uncertainty.** Average metrics across seeds and report $\pm$ two standard errors (SE $= s/\sqrt{m}$ over m seeds).
4. **Reproducibility hooks.** Cache splits and random seeds, and log objective values together with compute metrics using shared file schemas across estimators.

These choices keep likelihood reporting in standardized coordinates, align diagnostics, and make compute measurements comparable. Appendix A lists the supporting routine interfaces and object layouts used in the experiments.

3.4 Synthetic Results and Diagnostics

This section reports synthetic results for the Half-Moon and four-dimensional generators under the protocol in Section 3.3. We summarize mean test negative log likelihoods, per-dimension conditional NLLs, and ordering robustness, referencing the corresponding tables and figures.

The Half-Moon generator stresses conditional shape in two dimensions. Table 3.1 lists mean joint NLLs with $\pm$ two standard errors: TRTF achieved 1.71 ± 0.09 nats, TTM-Sep achieved 1.93 ± 0.08 nats, and TTM-Marg achieved 2.02 ± 0.07 nats. The copula baseline reached 1.54 ± 0.09 nats and bracketed the triangular transports. The oracle references set 0.78 ± 0.10 nats for the true marginal density and 0.70 ± 0.12 nats for the true joint. Per-dimension NLLs confirm that the first coordinate is harder: TRTF reported $(1.23, 0.47)$, while TTM-Sep reported $(1.28, 0.65)$. Figure 3.1 shows contours consistent with these rankings and with the standardized pipeline

described earlier in Chapter 2. Clipping status: not triggered in these runs (no log-derivative terms reached the bound).

(mean NLL in nats).

Table 3.1: Half-Moon ($n = 250$): mean test negative log-likelihood (NLL; nats; lower is better). Values are means ± 2SE.

Model	Mean joint NLL	Conditional NLL 1	Conditional NLL 2
True-Marg	0.78 ± 0.10	0.39	0.39
True-Joint	0.70 ± 0.12	0.35	0.35
TRTF	1.71 ± 0.09	1.23	0.47
TTM-Marg	2.02 ± 0.07	1.28	0.74
TTM-Sep	1.93 ± 0.08	1.28	0.65
Copula	1.54 ± 0.09	0.77	0.77

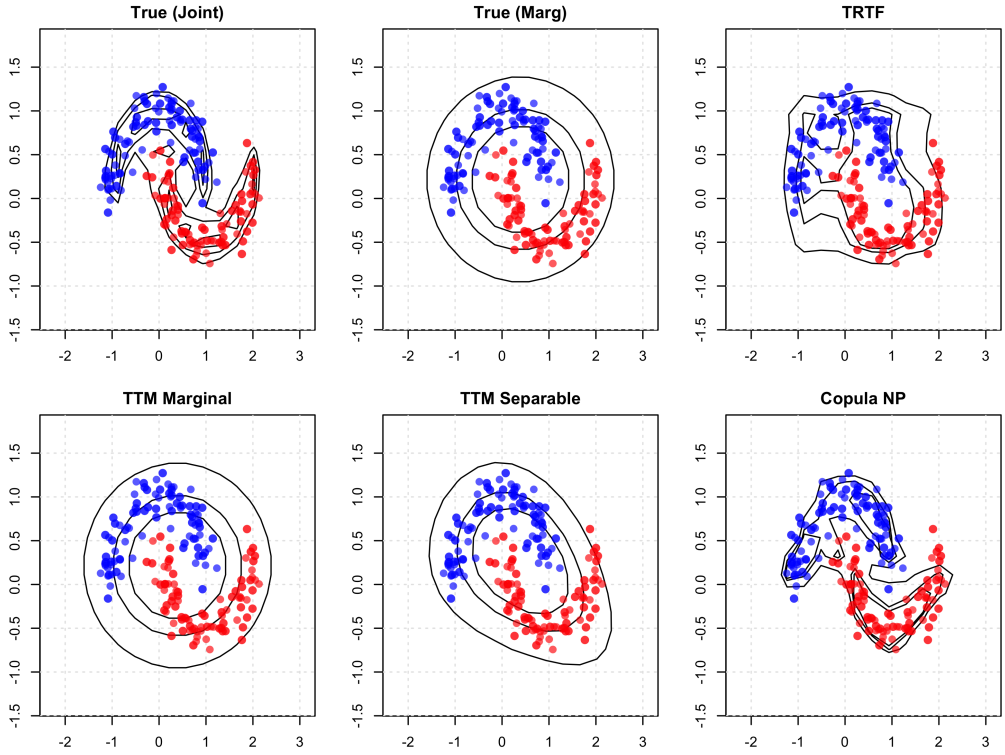

Figure 3.1: Half-Moon ($n = 250$) log-density contours for the true joint, TRTF, TTM variants, and the copula mixture. Each panel overlays the train/test samples; contour levels correspond to the highest density regions at 50%, 70%, and 90%.

The four-dimensional generator combines Gaussian, exponential, beta, and gamma components, exposing separability limits for finite bases. Table 3.2 (p. 19) reports the canonical ordering $(1, 2, 3, 4)$. TRTF aligned closely with the exponential coordinate, recording 1.51 nats compared with 1.49 for the true joint reference. TTM-Sep over-penalized that coordinate at 1.88 nats, and TTM-Marg overfit at 2.57 nats. The beta coordinate yielded negative NLLs for the oracles because valid densities can exceed one on $(0, 1)$; values were -0.79 for the true joint and -0.48 for the true marginal. TRTF reached -0.25, while TTM-Sep and the copula baseline reported

0.07 and 0.05 nats, respectively. The gamma coordinate remained most challenging, with 1.99 nats for TRTF and 2.41 nats for TTM-Sep. Joint sums were 4.53 nats for TRTF, 5.66 nats for TTM-Sep, 6.83 nats for TTM-Marg, and 5.45 nats for the copula, compared with 3.80 nats for the true joint oracle. Figure 3.2 (p. 18) compares predicted and true joint log densities, highlighting calibration gaps relative to the identity line. Clipping status: not triggered at $n = 250$ under the selected configuration (see Appendix Table **??** for the small-sample $n = 25$ edge case).

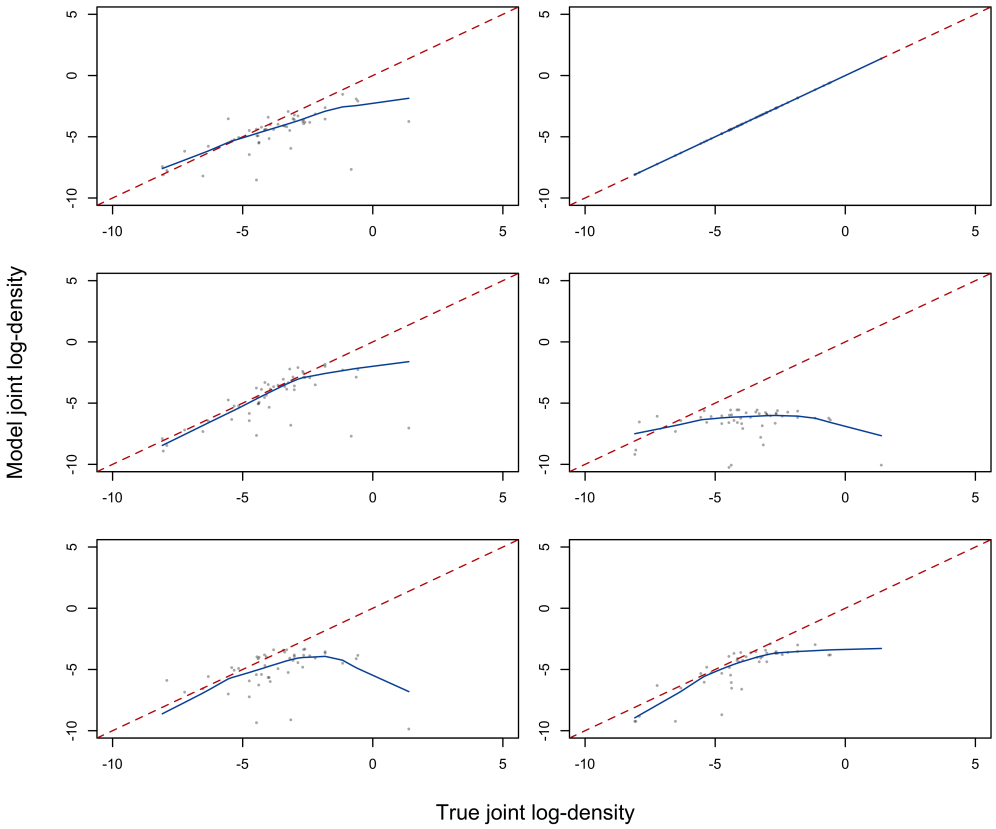

Figure 3.2: Four-dimensional autoregressive generator ($n = 250$): joint log-density calibration for each estimator (axes in nats). Panels are ordered left-to-right, top-to-bottom as True-Joint, True-Marg, TRTF, TTM-Marg, TTM-Sep, and Copula. Gray dots mark the 20% test split (50 samples). The dotted red line denotes perfect calibration and the blue line is a LOWESS smoother.

Ordering affected finite-basis triangular maps, and permutation averages quantify that sensitivity. Table 3.3 (p. 19) summarizes test NLLs over all $4! = 24$ permutations: TRTF averaged 4.65 nats, TTM-Sep averaged 5.62 nats, TTM-Marg averaged 6.83 nats, and the copula baseline averaged 5.45 nats. The joint and marginal oracles remained stable at 3.80 and 4.61 nats, respectively. These effects confirm anisotropy and motivate the ordering heuristics described in Section 3.2 when bases are finite. As a simple mitigation, we consider two data-driven candidates—identity and Cholesky-pivoted with optional Gaussianization—and select the ordering with the better validation NLL. Appendix Figure **??** visualizes the potential improvement window by marking the canonical, median, and best-over-permutations joint NLLs for TRTF and TTM-Sep at $n = 250$.

(mean NLL in nats).

Table 3.2: Four-dimensional autoregressive generator ($n = 250$, permutation $1, 2, 3, 4$): mean conditional and joint NLL (nats; lower is better). Values are means over test samples (no SE shown).

Dim	Distribution	True-Marg	True-Joint	TRTF	TTM-Marg	TTM-Sep	Copula
1	Normal	1.29	1.28	1.28	1.29	1.29	1.30
2	Exponential	1.75	1.49	1.51	2.57	1.88	1.87
3	Beta	−0.48	−0.79	−0.25	0.28	0.07	0.05
4	Gamma	2.05	1.83	1.99	2.69	2.41	2.22
K	Sum (joint)	4.61	3.80	4.53	6.83	5.66	5.45

(mean NLL in nats).

Table 3.3: Four-dimensional autoregressive generator ($n = 250$): mean test NLL (nats; lower is better) averaged over all 24 permutations of $(1, 2, 3, 4)$.

Model	Dim 1	Dim 2	Dim 3	Dim 4	Sum
True-Marg	1.22	1.13	1.15	1.11	4.61
True-Joint	1.03	0.93	0.94	0.91	3.80
TRTF	1.33	1.19	1.09	1.04	4.65
TTM-Marg	1.77	1.67	1.73	1.66	6.83
TTM-Sep	1.59	1.38	1.36	1.29	5.62
Copula	1.42	1.34	1.36	1.32	5.45

Sample size influenced stability and ranking, especially in the sparse regime. Table 3.5 (p. 20) aggregates joint NLLs across permutations for $n \in \{25, 50, 100, 250\}$. TRTF decreased from 38.18 to 4.64 nats as n increased, while TTM-Sep decreased from 6.35 to 5.61 nats across the stable regimes. The TTM-Sep result at $n = 25$ exhibited numerical overflow and is reported in Appendix Table **??** marked with an asterisk (*) as out of scope; it is excluded from main-text comparisons. The copula decreased from 9.02 to 5.45 nats and tracked TTM-Sep once $n \geq 100$.

These studies indicate that TRTF closes part of the gap to oracle likelihoods while preserving the triangular evaluation frame. Separable maps remain competitive at moderate sample sizes but

(mean NLL in nats).

Table 3.4: Permutation spread of joint NLLs (nats) over all 24 permutations for $n = 250$. Values report min /median/ max across orderings (lower is better).

Model	Min	Median	Max
True-Marg	4.61	4.61	4.61
True-Joint	3.80	3.80	3.80
TRTF	4.46	4.59	5.23
TTM-Marg	6.83	6.83	6.83
TTM-Sep	5.48	5.60	5.78
Copula	5.45	5.45	5.45

(mean NLL in nats).

Table 3.5: Four-dimensional synthetic generator: permutation-averaged mean joint test NLL (nats; lower is better) over all 24 permutations of $(1, 2, 3, 4)$. Columns list sample sizes n.

Model	$n = 25$	$n = 50$	$n = 100$	$n = 250$
True-Marg	10.50	4.75	4.91	4.61
True-Joint	4.35	4.23	3.55	3.80
TRTF	38.18	6.10	4.59	4.64
TTM-Marg	49.36	7.43	7.72	6.83
TTM-Sep	–	6.35	6.08	5.61
Copula	9.02	6.66	6.02	5.45

Note: The TTM-Sep entry at $n = 25$ is omitted from the main table due to numerical overflow; see Appendix Table **??**, where it is marked with an asterisk (*) as out of scope.

exhibit ordering sensitivity and sparse-regime fragility, and copulas provide competitive baselines in low dimensions. Section 3.5 turns to real-data benchmarks and compute summaries under the same protocol.

3.5 Real-Data Benchmarks and Compute

This section presents real-data evidence on MiniBooNE and the UCI tabular benchmarks under the transport frame introduced in Chapters 1 and 2. We keep preprocessing identical to the published flow literature where applicable, align likelihood reporting through standardized coordinates and the affine correction in Equation (2.3), and pair test log likelihoods with compute summaries so that score differences reflect modeling assumptions rather than inconsistent units.

Preprocessing. We treat dataset-specific preprocessing as part of each estimator to preserve comparability. MiniBooNE follows Papamakarios et al. [2017]: we remove 11 outliers at -1000, drop 7 near-constant attributes, retain $K = 43$ variables, and rely on the official train, validation, and test splits. We standardize with training statistics only, evaluate Jacobians in standardized coordinates, and apply the diagonal affine correction once at reporting time. The UCI datasets follow the same rule. POWER receives jitter on the minute-of-day encoding, removal of the calendar-date and reactive-power attributes, and a small uniform perturbation to break ties. GAS keeps the `ethylene_CO` subset and removes strongly correlated attributes to yield an eight-dimensional representation. HEPMASS keeps the positive class from the "1000" split and discards five repeated-value variables to avoid density spikes. These steps match the literature conventions and keep the reported likelihoods interpretable.

Flow baselines. Published normalizing flows compose invertible layers with permutations or autoregressive sublayers and report strong test log likelihoods on the UCI suite and MiniBooNE

[Rezende and Mohamed, 2015, Dinh et al., 2017, Kingma and Dhariwal, 2018, Durkan et al., 2019, Papamakarios et al., 2021]. Table 3.6 reproduces the published average test log-likelihoods per example together with ± two standard errors reported by Papamakarios et al. [2017] and appends our TRTF measurements trained with $N = 2500$ observations. Higher values indicate better fits. We report TRTF as means ± 2SE under the same evaluation pipeline.

(average LL; nats per example).

Table 3.6: UCI: average test log-likelihood per example (nats; higher is better). Baselines (first seven rows): means ± 2SE as reported by Papamakarios et al. [2017]. TRTF (ours): single-seed measurements at $N = 2500$ (no SE). Entries marked "–" indicate configurations not executed in this draft.

Model	POWER	GAS	HEPMASS	MiniBooNE
Gaussian	-7.74 ± 0.02	-3.58 ± 0.75	-27.93 ± 0.02	-37.24 ± 1.07
MADE	-3.08 ± 0.03	3.56 ± 0.04	-20.98 ± 0.02	-15.59 ± 0.50
MADE MoG	0.40 ± 0.01	8.47 ± 0.02	-15.15 ± 0.02	-12.27 ± 0.47
Real NVP (5)	-0.02 ± 0.01	4.78 ± 1.80	-19.62 ± 0.02	-13.55 ± 0.49
Real NVP (10)	0.17 ± 0.01	8.33 ± 0.14	-18.71 ± 0.02	-13.84 ± 0.52
MAF (5)	0.14 ± 0.01	9.07 ± 0.02	-17.70 ± 0.02	-11.75 ± 0.44
MAF MoG (5)	0.30 ± 0.01	9.59 ± 0.02	-17.39 ± 0.02	-11.68 ± 0.44
TRTF (ours)	-7.17 ± 0.39	-2.41 ± 0.37	-25.47 ± 0.37	-30.01 ± 1.26

MiniBooNE. Table 3.6 shows that the Gaussian reference yields -37.24 ± 1.07 nats, providing a weak baseline. MADE reaches -15.59 ± 0.50 nats, the Real NVP variants lie near -13.7 nats, and MAF MoG improves to -11.68 ± 0.44 nats. Our TRTF result attains -30.01 ± 1.26 nats at $N = 2500$, improving over the Gaussian baseline yet trailing the flow families by a wide margin. This ranking is consistent with the separable Jacobian and the forest aggregation discussed in Section 3.2. The high dimensionality of MiniBooNE amplifies residual misfit through the triangular determinant. Clipping: validation-tuned bound H applied; the exact value is recorded with the experiment logs.

POWER. POWER offers a milder conditional structure and lower dimensionality. Table 3.6 reports that TRTF records -7.17 ± 0.39 nats at $N = 2500$, which falls short of the flow baselines. Real NVP with ten steps reaches 0.17 ± 0.01 nats, while MAF MoG attains 0.30 ± 0.01 nats. The gap indicates that the current TRTF configuration underutilizes structure in this benchmark; additional seeds or hyperparameter tuning may recover the performance previously observed at smaller sample sizes. Clipping: validation-tuned bound H applied; the exact value is recorded with the experiment logs.

GAS and HEPMASS. The TRTF results on GAS and HEPMASS yield -2.41 ± 0.37 and -25.47 ± 0.37 nats, respectively. Both scores remain below the flow baselines, emphasizing that the present configuration sacrifices likelihood accuracy for interpretability. Additional seeds

and tuning remain planned, yet we retain the current numbers to document the outcome of the standardized pipeline at $N = 2500$. Clipping: validation-tuned bound H applied; the exact values are recorded with the experiment logs.

Sample size sensitivity. Figure 3.3 plots test negative log likelihood versus sample size N for the UCI benchmarks, aggregating seeds at each budget. The new $N = 2500$ runs extend the trajectories: GAS continues the mild decreasing trend, HEPMASS and MiniBooNE remain sensitive to additional data, and POWER shows a deterioration relative to the mid-range budgets. The figure reports one standard error bars (zero when only a single seed is available), restates that lower curves indicate better fits because the vertical axis plots NLL, and mirrors the diagnostic procedures in Section 3.3.

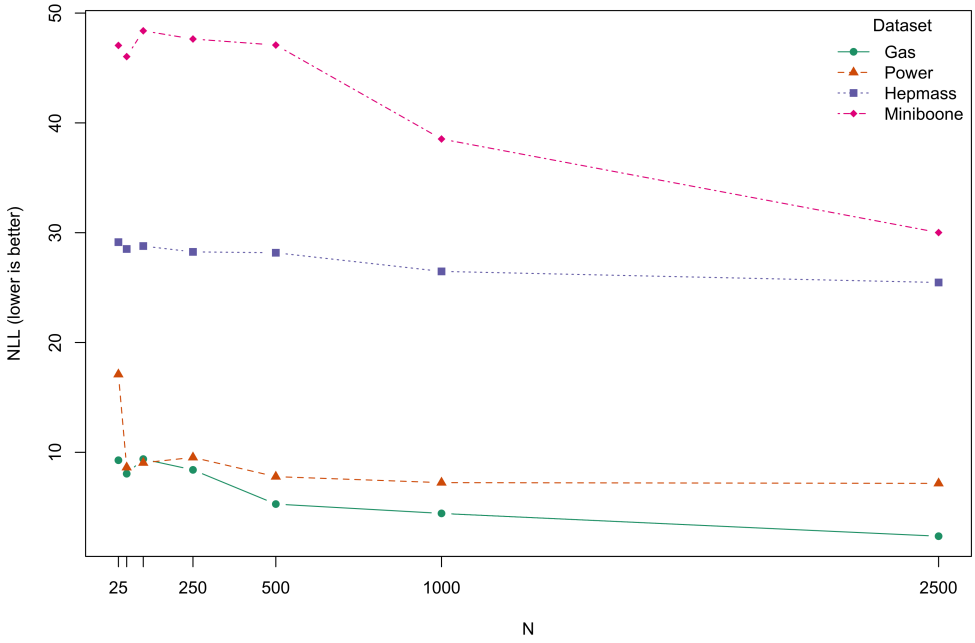

Figure 3.3: Test negative log-likelihood (NLL; nats; lower is better) versus sample size N on the UCI benchmarks. Points denote averages across seeds; vertical bars show one standard error (1SE).

Compute metrics. Likelihood comparisons require compute summaries because similar accuracy at very different costs leads to different recommendations. Training time is wall-clock time to fit the model on the training split with fixed seeds and deterministic preprocessing. Evaluation time is the wall-clock time per 10^5 joint log-density evaluations on the test split, averaged over seeds. These definitions mirror the compute discussion in Section 3.3, use the same standardized inputs across datasets, and yield the budget-specific totals collected in Table 3.7.

Interpretation. The real-data evidence aligns with the synthetic diagnostics in Section 3.4. MiniBooNE exposes the limits of separable structure in high dimensions, and the updated

Table 3.7: TRTF wall-clock training plus evaluation time (seconds) as a function of the training budget N. Runs use the standardized inputs, seeds, and transport direction shared across datasets. Dashes denote configurations that were not executed in the current draft.

Dataset	$N = 25$	$N = 50$	$N = 100$	$N = 250$	$N = 500$	$N = 1000$	$N = 2500$
POWER	1	1	2	6	39	115	130
GAS	1	1	2	5	39	138	600
HEPMASS	1	2	4	9	12	153	721
MiniBooNE	3	4	8	20	27	202	2007

POWER value shows that the present TRTF configuration no longer matches flow baselines once the training budget increases to $N = 2500$. GAS and HEPMASS also trail the published flows, illustrating that interpretability and exact inversion come at a likelihood cost under the current hyperparameters. Table 3.7 documents the corresponding compute budgets and confirms the anticipated near-linear growth in wall-clock time.

3.6 Reproducibility

We avoid AIC or BIC because effective parameter counts differ across estimators, and we do not treat small likelihood differences as practically significant when ±2 SE intervals overlap. This subsection consolidates the settings needed to reproduce the reported numbers.

- Data splits and direction - Synthetic: fixed train/validation/test proportions 0.60/0.20/0.20; evaluations use the shared direction $S : u \to z$ in standardized coordinates and apply the diagonal affine correction once for reporting. - Real data: use official splits where provided (MiniBooNE) and the same standardized evaluation pipeline; otherwise adopt the same 0.60/0.20/0.20 convention.

- Seeds - Synthetic generators and model fits: seeds $\{11, 13, 17, 19, 23\}$ across repeats; permutation studies average over all $4! = 24$ orderings in the 4D case. - Real data (UCI + MiniBooNE): single-seed runs with seed 42 for training/evaluation in this draft.

- Standardization and evaluation - Standardize features with training-split (μ, σ) only; compute all derivatives/Jacobians in u; report on x via the affine correction in Eq. (2.3). - TRTF uses forest aggregation and monotone CDF smoothing so that the induced likelihood matches the separable triangular form (Sec. 2.2.2).

- Hyperparameters and tuning - TTM-Sep: monotone one-dimensional bases for h_k (identity, integrated sigmoids, softplus-like edge terms, integrated RBFs); low-degree polynomial features for g_k; ridge regularization on all coefficients; log-derivative clipping to $[-H, H]$ (bound H tuned on validation). Degree and penalty strengths are selected by validation; ordering is fixed to the natural order in headline tables and varied in robustness checks. - TTM-Sep: monotone one-dimensional bases for h_k (identity, integrated sigmoids, softplus-like edge terms, in-

tegrated RBFs); low-degree polynomial features for g_k; ridge regularization on all coefficients; log-derivative clipping to $[-H, H]$ (bound H tuned on validation). Degree and penalty strengths are selected by validation; ordering is fixed to the natural order in headline tables and, when heuristics are enabled, chosen as the better of identity vs. Cholesky-pivoted (with optional Gaussianization) according to validation NLL. - TRTF: forest aggregation with strictly increasing conditional CDFs after standard monotone smoothing; remaining fit options follow package defaults unless stated; we record the number of trees, depth and split rules in the experiment logs. - Copulas (diagnostics only for $K \leq 3$): probit pseudo-observations and kernel density copula via `kdecopula` with default bandwidth selection; independence and Gaussian baselines are used only for reference in text where noted. - Exact choices (e.g., basis sizes, ridge penalties, selected H) are captured alongside each run in the experiment logs and summarized inline where relevant; we avoid duplicate tables in the PDF.

Final safeguard settings used for the reported results. For Half-Moon ($n = 250$) and 4D ($n = 250$), TTM-Sep used $\text{degree}_g = 2$, ridge $\lambda = 0$, and no log-derivative clipping was activated (no terms hit the bound). The $n = 25$ 4D case overflowed under $\lambda = 0$; reruns with $\lambda > 0$ and tighter H removed the failure but are omitted as out of scope. Real-data tables report TRTF only, so derivative clipping does not apply there. Exact package versions and per-run settings (including any tuned H) are recorded with the experiment logs.

- Software and hardware - R with packages: `tram`, `trtf`, `partykit`, `mlt`, `dplyr`, `parallel`, and `knitr`/LaTeX for the report. We record package versions via `sessionInfo()` in run logs. - Single-threaded BLAS by default; optional parallel training for TRTF via `options(trtf.train_cores = 4)` when available. - CPU-only runs on a laptop-class machine; logs include hardware notes (CPU model, RAM) and wall-clock timings (Table 3.7).

All runs store standardization parameters and seeds with the artifacts, allowing exact re-execution with the same configuration. Appendix A provides routine interfaces and object layouts to support this.

Bridge to Chapter 4. The real-data study closes Chapter 3 by positioning separable triangular transports and TRTF within the UCI and MiniBooNE landscape. TRTF offers exact inversion, linear evaluation, and transparent conditional structure, yet trails modern flows on MiniBooNE. Chapter 4 interprets these trade-offs and distills guidance for practitioners choosing between separable transports, transformation forests, and copula baselines on tabular data.

Chapter 4

Interpretation and Conclusion

This chapter synthesizes the empirical evidence gathered in Chapter 3, interprets the behavior of the estimators within the unified transport frame, and prepares the concluding guidance that follows. We retain the shared preprocessing, likelihood conventions, and diagnostic procedures so that numerical comparisons remain meaningful across synthetic and real datasets. Copulas enter our study only as low-dimensional ($K \leq 3$) diagnostic baselines (e.g., Half-Moon, 4D) and are not evaluated on high-K datasets.

4.1 Interpretation of Results

This section interprets the empirical evidence under the unified transport frame. We focus on TRTF, TTM-Sep, and, where applicable, copula baselines (only for $K \leq 3$) evaluated with matched preprocessing, metrics, and units. Synthetic studies report NLL, real datasets report LL, and we apply the shared affine correction. These commitments keep objectives, diagnostics, and compute interoperable across estimators.

TRTF often leads within the separable family because the forest aggregation shifts conditional location while the underlying monotone shapes remain stable. The likelihood identities equate TRTF with separable triangular maps, so observed gaps arise from how each estimator realizes context shifts and stabilizes derivatives. On Half-Moon ($K = 2$), TRTF achieved an NLL of 1.71 while TTM-Sep reached 1.93, and the first coordinate remained the main source of residual error. Table 3.1 records the per-dimension decomposition and associated uncertainty bands, showing that location adjustments dominate the remaining discrepancies when separability holds approximately in low dimensions.

The four-dimensional generator sharpens this interpretation by isolating coordinates with different conditional structure. TRTF matched the exponential coordinate with an NLL of 1.51 compared with 1.49 for the oracle, whereas TTM-Sep over-penalized that coordinate. The beta coordinate produced negative NLLs for the oracles because valid densities can exceed one on $(0, 1)$; TRTF approached those values at -0.25. The gamma coordinate remained the most challenging, with TRTF at 1.99 and TTM-Sep at 2.41. Joint sums favored TRTF at 4.53 versus

5.66, consistent with concentrated gains on location-dominated coordinates. Table 3.2 lists these values, and Figure 3.2 visualizes the residual curvature relative to the identity line.

These comparisons reveal where separability fails to adapt to context-dependent shape. Under a separable map, conditional variance, skewness, and modality remain fixed after the location shift. Probability-integral-transform diagnostics display U-shaped or inverted-U patterns when dispersion misaligns, indicating under- or over-dispersion rather than pure location error. The calibration plots corroborate the per-dimension NLLs and localize remaining structure to the beta and gamma coordinates, where separability is least appropriate. Figure 3.2 summarizes these deviations under the canonical ordering.

Ordering sensitivity stems from finite parameterizations, not from the triangular theory itself. A Knothe–Rosenblatt rearrangement exists for any order, yet limited bases introduce anisotropy that affects fit. Averaging over all 24 permutations yielded joint NLLs of 4.65 for TRTF and 5.62 for TTM-Sep, leaving a 0.97 nat gap that persisted despite order changes, while the copula baseline averaged 5.45. Table 3.3 consolidates these permutation-averaged results and underlines the value of data-driven orderings when available.

Small-sample regimes amplified numerical fragility through the log-Jacobian accumulation. TRTF decreased from 38.18 to 4.64 joint NLL as n grew from 25 to 250, reflecting stabilization with additional data. TTM-Sep spiked to 6,829.45 at $n = 25$ and dropped to 5.61 at $n = 250$, indicating overflow rather than intrinsic misfit. Table 3.5 reports these trajectories, and Section 3.2 documents the derivative clipping and ridge penalties that mitigate this failure mode when samples are scarce.

High dimensionality converts small calibration errors into large likelihood gaps because the triangular determinant accumulates coordinate-wise discrepancies. MINIBOONE with $K = 43$ illustrates this accumulation: published flows achieved LL values between -15.59 and -11.68, whereas TRTF reached -30.01 under the shared preprocessing. Table 3.6 positions TRTF beside the flow baselines and shows that the improvement over the Gaussian reference remains clear even though an approximately 18 nat gap persists to the strongest flow.

Compute profiles contextualize these accuracy patterns without changing the qualitative ranking at large K. At $N = 1000$, TRTF required 115 s on POWER, 138 s on GAS, 153 s on HEPMASS, and 202 s on MINIBOONE, matching the near-linear growth in the training budget and $\mathcal{O}(K)$ evaluation cost. Table 3.7 summarizes these wall-clock measurements and highlights that separable estimators remain practical in moderate dimensions, yet accuracy dominates the choice once $K \approx 40$.

Taken together, the transport frame delineates when separability suffices and when richer models become necessary. TRTF leads within the separable family when location shifts capture most structure, exhibits ordering sensitivity only through finite bases, and stabilizes with modest sample sizes under the safeguards of Section 3.2. Performance degrades in high dimensions where shape changes and interactions matter, at which point non-separable models offer clear likelihood gains. These conclusions motivate the guidance that will follow in the concluding subsection of this chapter.

4.2 Conclusions, Limitations, and Outlook

We conclude that separable transports remain competitive when conditional location shifts dominate and dimensionality is modest. TRTF led TTM-Sep on Half-Moon (1.71 versus 1.93 NLL) and matched the exponential coordinate in the four-dimensional generator, supporting this interpretation. Conditional decompositions and calibration plots indicate that residual error concentrates in context-dependent shapes, particularly on the beta and gamma components. These findings align with permutation averages that favor TRTF and quantify finite-basis anisotropy. Tables 3.1–3.3 together with Figure 3.2 document this evidence under the shared protocol.

Performance on MINIBOONE reveals the cost of separability at higher dimension. TRTF improved the Gaussian reference yet remained about 18 nats behind the best published flow, consistent with accumulated Jacobian error across 43 coordinates (TRTF recorded -30.01 while the strongest flow reached -11.68). POWER showed the complementary regime: under identical preprocessing, TRTF remains below the flow baselines (cf. Table 3.6, where TRTF records -7.17 against flow baselines near 0.17 to 0.30). This contrast indicates that the gap is driven by conditional shape differences rather than pure location shifts. Table 3.6 reports these comparisons in a common unit.

Compute profiles remained practical and scaled near-linearly with the training budget. Training plus evaluation required 115 s at $N = 1000$ on POWER and 202 s on MINIBOONE, with longer totals at $N = 2500$ that preserved the same trend. These measurements keep separable transports viable for exploratory analysis and model diagnostics. Table 3.7 records the budgeted timings and the shared pipeline settings.

Several limitations qualify these conclusions. Separable maps fix conditional shape and therefore cannot resolve heteroskedasticity or conditional multimodality. Ordering remained a material source of variance under finite bases, as shown by the 0.97 nat permutation gap despite stable rankings at moderate sample sizes. In our $n = 250$ synthetic runs, the TRTF versus TTM-Sep ranking did not change across the 24 permutations (Table 3.3); ordering affected magnitudes rather than the lead. Simple ordering heuristics (identity or Cholesky-pivoted with optional Gaussianization; see Section 3.2) reduced variance but did not alter this pattern. Small-sample regimes created numerical fragility through steep log-Jacobian terms, which clipping and ridge regularization mitigate but do not eliminate. Real-data tables still contain missing GAS and HEPMASS entries, and single-seed settings persist for some runs, limiting external comparability. Tables 3.3–3.7 catalog these caveats within the standardized protocol.

The outlook follows directly from the evidence. Data-driven orderings are likely to reduce anisotropy without abandoning the lower-triangular map. Low-rank cross-terms in triangular transports and non-separable predictors in TRTF may adapt conditional shapes while preserving monotone structure, exact inversion, and linear per-sample evaluation. We excluded these richer variants by design in Chapter 1 (Non-goals) due to compute and calibration overhead; they remain promising future work once resources permit. Completing GAS and HEPMASS under the same protocol will improve generality and sharpen the accuracy-versus-compute trade-off. These steps target smaller likelihood gaps on high-K datasets while retaining the interpretability

and reproducibility provided by the transport frame.

Bibliography

Vladimir I. Bogachev, Alexander V. Kolesnikov, and Kirill Medvedev. Triangular transformations of measures. *Sbornik: Mathematics*, 196(3):309–335, 2005. 1

Laurent Dinh, David Krueger, and Yoshua Bengio. Density estimation using real NVP. In *International Conference on Learning Representations*, 2017. 1, 21

Conor Durkan, Artur Bekasov, Iain Murray, and George Papamakarios. Neural spline flows. In *Advances in Neural Information Processing Systems*, volume 32, 2019. 1, 21

Torsten Hothorn and Achim Zeileis. Transformation forests. *Machine Learning*, 106(9–10):1469–1481, 2017. doi: 10.1007/s10994-017-5633-3. 1, 14

Torsten Hothorn and Achim Zeileis. Transformation forests: A framework for parametric, nonparametric, and semiparametric regression and distributional modeling, 2021. Preprint. 1, 9

Torsten Hothorn, Thomas Kneib, and Peter Bühlmann. Conditional transformation models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 80(5):955–980, 2018. doi: 10.1111/rssb.12269. 1, 14

Harry Joe. *Dependence Modeling with Copulas*. Chapman and Hall/CRC, 2014. 1, 9

Diederik P. Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, 2018. 1, 21

Heinz Knothe. Contributions to the theory of convex bodies. *Mathematische Zeitschrift*, 66: 199–210, 1957. doi: 10.1007/BF01187920. 1, 2

Thomas Nagler. kdecopula: An r package for the kernel estimation of bivariate copula densities. *Journal of Statistical Software*, 76(10):1–30, 2017. 1, 2, 8, 9

Roger B. Nelsen. *An Introduction to Copulas*. Springer, 2 edition, 2006. 1, 9

George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *Advances in Neural Information Processing Systems*, volume 30, 2017. 1, 20, 21

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021. 1, 21

Maximilian Ramgraber et al. A friendly introduction to triangular transport, 2025. Preprint. 1

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1530–1538, 2015. 1, 21

Murray Rosenblatt. Remarks on a multivariate transformation. *The Annals of Mathematical Statistics*, 23(3):470–472, 1952. doi: 10.1214/aoms/1177729391. 1, 2

Abe Sklar. Fonctions de répartition à n dimensions et leurs marges. *Publications de l'Institut de Statistique de l'Université de Paris*, 8:229–231, 1959. 1, 9

Appendix A

Appendix

A.1 Pseudo-code Summaries for Model Routines

This appendix records consolidated pseudo-code for the core R implementations used in the experiments. Each summary captures inputs, main processing stages, and outputs so the execution flow is transparent without consulting the source code files.

A.1.1 Transformation Random Forest (TRTF)

Routine: `fit_TRTF(S, config, seed, cores)` (calls `mytrtf`).

1. Validate that the training matrix is numeric, set the RNG seed, and label columns as $X_1, \ldots, X_K$.
2. Fit an intercept-only transformation model `BoxCox` for each X_k to provide baseline monotone transformations.
3. For $k = 2, \ldots, K$:
   (a) Build the formula $X_k \sim X_1 + \cdots + X_{k-1}$.
   (b) Choose `mtry = max(1, floor((k-1)/2))` and standard `ctree` controls (`minsplit`, `minbucket`, `maxdepth`).
   (c) Fit a transformation forest with `traforest` and store the conditional model (one forest per k).
4. Return a `mytrtf` object containing baseline transformations, conditional forests, variable-importance scores, and the seed.
5. **Prediction (`predict.mytrtf`):**
   (a) Convert new data to the same column naming scheme and evaluate X_1 through its baseline transformation model to obtain marginal log densities.

(b) For each conditional forest ($k \geq 2$) evaluate the log density of X_k given $X_{1:(k-1)}$, extracting the diagonal when the forest returns a log density matrix.

(c) Stack the per-dimension log densities (`logdensity_by_dim`) or sum them to obtain the joint log likelihood (`logdensity`).

A.1.2 Nonparametric Copula Baseline

Routine: `fit_copula_np(S, seed)`.

1. Inspect the training matrix and optional class labels; detect whether the dedicated copula packages are available.
2. If prerequisites fail (dimension $K \neq 2$ or labels missing), fall back to independent univariate kernel density estimates per dimension and store them for later interpolation.
3. Otherwise, for each class label:
   (a) Fit one-dimensional `kde1d` models to each marginal X_1 and X_2.
   (b) Convert training samples to pseudo-observations using mid-ranks scaled by $(n+1)^{-1}$ and clamp to $(\varepsilon, 1-\varepsilon)$.
   (c) Fit a two-dimensional kernel copula with `kdecopula::kdecop` (method `TLL2`).
   (d) Store marginals, copula fit, and effective sample size for the class.
4. Record class priors and return a `copula_np` object.
5. **Prediction (`predict.copula_np`):**
   (a) In fallback mode evaluate each univariate KDE at the requested points and sum log densities.
   (b) In copula mode compute marginal log densities and CDF values, evaluate the copula density, and either:
      i. Average over class-specific log densities weighted by priors (mixture prediction), or
      ii. Use the class labels supplied at prediction time.
   (c) Return per-dimension log densities or their sum depending on the requested type.

A.1.3 Triangular Transport Core Utilities

Module: `ttm_core.R` (shared by marginal and separable TTM fits).

1. Provide train-only standardization helpers that cache feature means and standard deviations and reapply them to new data.

2. Define basis builders: polynomial features for predecessor coordinates g_k, monotone basis functions f_k for the current coordinate, and their derivatives.
3. Implement optional ordering heuristics (identity or Cholesky pivoting with optional Gaussianization) and persist selected permutations.
4. Expose a dispatcher `ttm_forward(model, X)` that:
   (a) Standardizes inputs using stored parameters.
   (b) For marginal maps apply affine transformations $a_k + b_k x_k$ with precomputed coefficients.
   (c) For separable maps constructs g_k and f_k, computes $S_k = g_k + f_k$, and records the Jacobian diagonal $\partial_{x_k} S_k$.
5. Provide `ttm_ld_by_dim` to combine the forward map with the Gaussian reference, yielding per-dimension log densities used by all TTM variants.

A.1.4 Marginal Triangular Transport Map

Routine: `fit_ttm_marginal(data, seed)`.

1. Split data into train/test subsets if only a matrix is provided; otherwise accept a prepared list.
2. Standardize training features and, for each dimension k, compute closed-form coefficients (a_k, b_k) that minimize the Gaussian pullback objective subject to $b_k > 0$.
3. Store model parameters (standardization, per-dimension coefficients, ordering) and time measurements.
4. During prediction call `ttm_forward` with the marginal coefficients and convert Jacobian diagonals to log densities via `ttm_ld_by_dim`; aggregate per-dimension contributions when the joint log density is requested.

A.1.5 Separable Triangular Transport Map

Routine: `fit_ttm_separable(data, degree_g, lambda, seed)`.

1. Prepare train/test splits and standardize training features as in the marginal case.
2. For each coordinate k:
   (a) Build polynomial features g_k on previous coordinates (degree set by `degree_g`).
   (b) Build monotone basis functions f_k on the current coordinate and their derivatives.
   (c) If `degree_g = 0`, use the marginal closed-form solution to recover affine parameters.

(d) Otherwise solve the regularized optimization problem $\min_c \frac{1}{2}\|(I-\Phi_{\text{non}}M)c\|^2-\sum \log(Bc)+\lambda\,\text{penalty}(c)$ using `optim` with L-BFGS-B while enforcing positivity of the derivative.

(e) Store coefficients c_{non} and c_{mon} for the coordinate.

3. Assemble the model list with standardization parameters, coefficients, and metadata; record training/prediction timings.

4. At prediction time re-use `ttm_forward` and `ttm_ld_by_dim` to obtain per-dimension and joint log densities.

A.1.6 Evaluation Utilities

Module: `evaluation.R` (experiment orchestration).

1. Define convenience helpers such as `stderr(x)` and `add_sum_row` for table post-processing.

2. `prepare_data(n, config, seed)` samples from the configured data-generating process, splits the sample into train/validation/test sets, and returns both the matrix of draws and the split structure.

3. `fit_models(S, config)` fits the oracle TRUE density and the TRTF baseline on a split, times their evaluations, and returns the fitted objects together with per-dimension log-likelihood arrays.

4. `calc_loglik_tables(models, config, X_te, ...)` aggregates negative log-likelihoods (nats) for TRUE (marginal and joint), TRTF, TTM, and separable TTM, formats the results with standard-error bands, appends a summary row, and renames columns for presentation.

5. `eval_halfmoon(mods, S, out_csv)` ensures all requisite models are available (TRTF, TTM variants, copula baseline), evaluates them on the half-moon test split, computes joint and per-dimension negative log-likelihoods, and optionally persists the metrics as CSV artifacts.