Multivariate Density Estimation: Comparing Transformation Random Forest, Normalizing Flows and Copulas

Léon Kia Faro

2025-09-02

Contents

1 Introduction

Estimating the joint density $p(x_1, \dots, x_K)$ of a multivariate random vector $x = (x_1, \dots, x_K)$ is a core problem in statistical modeling and unsupervised learning. By learning $p(x)$ from data, we enable a range of tasks including probabilistic inference, anomaly detection, data imputation, and generative modeling. However, high-dimensional density estimation is challenging due to the curse of dimensionality and complex dependency structures. This thesis investigates three modern approaches to multivariate density estimation – **Triangular Transport Maps (TTM)**, **Transformation Random Forests (TRTF)**, and **Copula Models** – comparing their methodologies and performance. We focus on how each method transforms or models a complex joint distribution by leveraging simpler components, and we evaluate them on benchmark data.

A unifying perspective for density estimation is **measure transport**, where one transforms a complicated **target distribution** π on $\mathbb{R}^K$ into a simpler **reference distribution** η (often a standard Gaussian or uniform). If $S : \mathbb{R}^K \to \mathbb{R}^K$ is an invertible transport map such that $z = S(x)$ has distribution η when $x \sim \pi$, then we can estimate π via learning S and use S^{-1} for sampling from π. **Normalizing flows** implement this idea using deep neural networks: they construct S as a composition of many invertible transformations with tractable Jacobians. *Triangular transport maps*, in contrast, impose a structured, *monotonic* form on S that offers theoretical and practical advantages. Meanwhile, **transformation forests** (TRTF) take a nonparametric ensemble approach, using tree-based models to estimate conditional distributions. Finally, **copula models** provide a statistically interpretable way to construct multivariate densities by separating marginal distributions from the dependency structure (the copula). In this work, we formulate each approach in a common framework, strictly adopting the notation and terminology of a recent tutorial on triangular transport maps. We then empirically compare their performance on a real dataset and discuss theoretical connections between the methods.

2 Research Questions

We address the following research questions:

- **Q1: Performance** – How do TTM, TRTF, and copula models compare in terms of goodness-of-fit for multivariate density estimation? We will evaluate models on held-out test data via log-likelihood and other metrics to determine which approach best captures the true distribution of complex, high-dimensional data.
- **Q2: Trade-offs** – What are the trade-offs in complexity, interpretability, and computational efficiency between the methods? For instance, triangular maps offer *exact* likelihood evaluation and have interpretable structure, while normalizing flows and forests might be more flexible but less transparent. We also ask if simpler parametric copulas suffer when assumptions are violated.

3 Methods

We outline the methodology of each approach. We begin with triangular transport maps (a type of normalizing flow with a structured form), then describe transformation random forests, copula models, and finally define baseline oracle models. Throughout, we let $\pi(x)$ denote the unknown target density of interest (on $\mathbb{R}^K$) and $\eta(z)$ a reference density (typically a standard multivariate normal or product of uniforms). All methods ultimately seek to estimate π by either constructing a map S such that $S(X) \sim \eta$ for $X \sim \pi$, or by directly modeling π via factorization into lower-dimensional components.

3.1 Triangular Transport Maps

A *triangular* map S is built component-wise, with each output S_k a function of the first k inputs only, and strictly increasing in x_k. Monotonicity ensures that S is invertible and that the Jacobian determinant factorizes as $\det \nabla S(x) = \prod_{k=1}^{K} \partial_{x_k} S_k(x_{1:k})$, making change-of-variables densities efficient to compute. Inverting S reveals the autoregressive factorization

$$\pi(x_1, \dots, x_K) = \pi(x_1)\,\pi(x_2 \mid x_1) \cdots \pi(x_K \mid x_{1:K-1}),$$

so each component S_k^{-1} acts as a conditional quantile. Parameterizations used in this work include marginal/diagonal, separable, and cross-term forms (see Section 3.1.0–3.1.1 above for implementation formulas).

3.2 Transformation Random Forests (TRTF)

TRTF estimates the joint via a chain of conditional models $\pi(x_k \mid x_{1:k-1})$, one forest per k. Each forest partitions the covariate space and fits a parametric distribution locally for the response. The ensemble yields a smooth estimate of the conditional CDF/PDF, which we sum in log-space to obtain the joint log-likelihood. This defines an implicit triangular CDF map $\hat{S}_k = \hat{F}_k(x_k \mid x_{1:k-1})$ that is monotone in x_k by construction, linking TRTF to triangular transport.

3.3 Copula Models (Parametric and Nonparametric)

Copulas decouple marginals from dependence: with $u_i = F_{X_i}(x_i)$ and copula density $c(u)$, the joint density factorizes as $\pi(x) = c(u) \prod_i f_{X_i}(x_i)$. We consider a Gaussian copula with empirical marginals (semiparametric) and a smoothed empirical copula (nonparametric). These highlight the cost of misspecified dependence (parametric) versus the data demands of flexible nonparametric dependence estimation.

3.4 Baselines: True Joint and True Marginals

For simulations, **True Joint** provides an oracle upper bound on test log-likelihood, while **True Marginals** (independent product of true marginals) isolates the contribution of dependence modeling. For real data (MiniBooNE), only the independent baseline is feasible.

4 Statistical Evaluation Framework

4.1 Data and Preprocessing

For MiniBooNE we follow standard preprocessing: train-only standardization, removal of near-constant attributes, and pruning highly correlated features (to avoid trivial ridges). Established train/val/test splits are used to ensure comparability with prior work.

4.2 Log-Likelihood Estimation

Primary metric is *test* log-likelihood in nats. TTM and copulas allow exact evaluation via change-of-variables and copula factorization, respectively. TRTF evaluates the sum of conditional pdf logs, using the local parametric family from each forest leaf. We also compute calibration checks (marginal QQ) and dependence diagnostics (rank correlations) as secondary assessments.

4.3 Fairness and Robustness

All methods share the same splits, seeds, and comparable complexity; each result is averaged over multiple seeds with standard errors reported. We apply paired tests or overlap of $\pm 2 \cdot \mathrm{SE}$ to gauge practical significance differences.

5 Results

5.1 Half-Moon

#	model	mean_joint_nll	per_dim_nll_1	per_dim_nll_2
1	True_uncond.	1.37 ± 0.11	0.69	0.69
2	True_cond.	0.70 ± 0.12	0.35	0.35
3	TRTF	1.83 ± 0.14	1.25	0.57
4	TTM_marginal	2.04 ± 0.12	1.29	0.75
5	TTM_sep	1.92 ± 0.14	1.29	0.64
6	TTM_cross	1.22 ± 0.20	0.92	0.29
7	Copula_np	0.87 ± 0.16	0.76	0.11

5.2 4D Conditional Data Generation

n = 50

		True	True	Random	Marginal	Separable	Cross-Term
Dim	Distribution	(marginal)	(joint)	Forest	Map	Map	Map
1	norm	1.46 ± 0.26	1.41 ± 0.29	1.48 ± 0.24	1.49 ± 0.20	1.46 ± 0.26	1.46 ± 0.25
2	exp	1.55 ± 0.46	1.38 ± 0.65	2.54 ± 0.75	3.30 ± 0.01	1.75 ± 0.70	2.58 ± 0.03
3	beta	−0.46 ± 0.63	−0.63 ± 1.00	−0.14 ± 0.34	0.40 ± 0.17	0.28 ± 0.70	0.39 ± 0.23
4	gamma	2.21 ± 1.11	2.07 ± 0.80	2.22 ± 1.08	2.78 ± 0.77	2.97 ± 1.45	3.00 ± 1.46
k	SUM	4.75 ± 1.10	4.23 ± 1.03	6.10 ± 1.61	7.97 ± 0.94	6.47 ± 1.92	7.43 ± 1.66

5.3 MINIBOONE Dataset

We trained TTM (monotone NN maps), TRTF (500 trees per conditional, depth 10), Gaussian and nonparametric copulas, and the independent baseline. Hyperparameters were selected by validation likelihood where applicable.

5.4 Benchmark Comparison on MINIBOONE

Average test log likelihood (in nats) for conditional density estimation. Error bars correspond to 2 standard deviations.

Model	**Miniboone**
Gaussian (indep. baseline)	-37.24 ± 1.07
MADE (ACN)	-15.59 ± 0.50
MADE MoG	-12.27 ± 0.47
Real NVP (5-layer)	-13.55 ± 0.49
Real NVP (10-layer)	-13.84 ± 0.52
MAF (5-layer)	-11.75 ± 0.44
MAF (10-layer)	-12.24 ± 0.45
MAF MoG (5-layer)	-11.68 ± 0.44
TRTF (Transformation Forest)	-29.88 ± 0.02 *(ours)*

5.5 Discussion of TRTF Result

TRTF substantially improves over the independent Gaussian baseline, indicating it learns nontrivial dependencies, but it underperforms modern flow models by a wide margin. Likely causes include bias from local parametric families, high-dimensional conditioning (43D), and the need for many partitions to capture complex interactions. Qualitatively, TRTF samples preserve first-order moments and some pairwise structures but miss higher-order structure, consistent with the likelihood gap.

5.6 Triangular Transport Map Results

5.7 Copula Model Results

5.8 True Joint/Marginal Baseline Results

6 Theoretical Links between TRTF and TTM

Let $\hat{F}_k$ be the TRTF estimate of $F_{X_k|X_{1:k-1}}$. Define $\hat{S}_k(x_{1:k}) = \hat{F}_k(x_k \mid x_{1:k-1})$. Then $\hat{\mathbf{S}}(x) = (\hat{S}_1, \dots, \hat{S}_K)$ is triangular and monotone in the last argument by construction, pushing the empirical distribution toward $\mathrm{Unif}(0,1)^K$. In the limit of infinite data and perfect conditional estimation, $\hat{\mathbf{S}}$ converges to the KR map. Thus, TRTF can be viewed as a nonparametric learner of triangular transport, providing a principled bridge between tree ensembles and measure transport.

7 Appendix A — Access confirmation & Mathematical pseudoalgorithms

Access confirmation. I have carefully read `/mnt/data/a_friendly_introduction_to_triangular_transport` and will strictly adhere to its notation (triangular maps $S = (S_1, \dots, S_K)$, change of variables, monotone S_k in the last argument, Jacobian product, forward-KL training, etc.) in what follows.

7.1 Common conventions (all models)

- Data: $X \in \mathbb{R}^{N\times K}$ split into train/val/test with a **global seed** s.
- Train-only standardization: for $k = 1, \dots, K$,

$$\mu_k = \tfrac{1}{N_{\text{tr}}}\sum_{i\in \text{tr}} x_{ik}, \qquad \sigma_k = \sqrt{\tfrac{1}{N_{\text{tr}}-1}\sum_{i\in\text{tr}}(x_{ik}-\mu_k)^2}, \quad \sigma_k > 0.$$

  Write $u_{ik} = (x_{ik} - \mu_k)/\sigma_k$ and $u = (x-\mu) \oslash \sigma$.
- API invariants:
  `predict`$(\cdot,$ “logdensity_by_dim”$) \to \mathbb{R}^{N\times K}$,
  `predict`$(\cdot,$ “logdensity”$) \to \mathbb{R}^{N}$ with row sums:

$$L_i = \sum_{k=1}^{K} \mathrm{LD}_{ik} \quad (\forall i), \qquad \text{no NA/Inf.}$$

- If a Gaussian reference $\eta = \mathcal{N}(0, I)$ is used via a triangular map $z = S(u)$, then for any x,

$$\boxed{\mathrm{LD}_k(x) = -\tfrac{1}{2} z_k(u)^2 - \tfrac{1}{2}\log(2\pi) + \log \partial_{u_k} S_k(u) - \log \sigma_k,}$$

  and $L(x) = \sum_k \mathrm{LD}_k(x)$.
  (Triangular structure: $S_k = S_k(u_{1:k})$ with $\partial_{u_k} S_k > 0$; Jacobian factorizes as $\det \nabla S = \prod_k \partial_{u_k} S_k$.)
- Determinism: identical seeds ⇒ identical outputs up to machine precision.

7.2 A) True Marginals (“oracle–independence” model)

Inputs. Oracle marginal pdfs $\{\pi_k\}_{k=1}^K$ (on original scale).

Fit. 1. Compute (μ, σ) on train; persist.

Predict. For each x (row-wise): 1. $u = (x - \mu) \oslash \sigma$. 2. For each k:

$$\pi_{k,\text{std}}(u_k) = \sigma_k\, \pi_k(\mu_k + \sigma_k u_k), \qquad \mathrm{LD}_k = \log \pi_{k,\text{std}}(u_k) - \log \sigma_k = \log \pi_k(x_k).$$

3. Output $\mathrm{LD} = (\mathrm{LD}_k)_{k=1}^K$, $L = \sum_k \mathrm{LD}_k$.

(Unit test: if $\pi_k = \mathcal{N}(0,1)$ and $\mu = 0, \sigma = 1$, then $\mathrm{LD}_k = -\tfrac{1}{2}x_k^2 - \tfrac{1}{2}\log(2\pi)$.)

7.3 B) True Joint ("oracle–autoreg." model)

Inputs. Either - (B1) oracle conditional pdfs $\{\pi(x_k \mid x_{1:k-1})\}_{k=1}^{K}$, or - (B2) only oracle joint pdf $\pi(x)$.

Fit. Compute (μ, σ) on train; persist.

Predict. For each x:

- **Case B1 (preferred; triangular factorization).**
  For $k = 1, \dots, K$, set

  $$\mathrm{LD}_1 = \log \pi(x_1), \qquad \mathrm{LD}_k = \log \pi(x_k \mid x_{1:k-1}) \quad (k \geq 2), \qquad L = \sum_k \mathrm{LD}_k = \log \pi(x).$$

  (Standardization is a no-op algebraically: $\log \pi(x) = \log \pi_{\mathrm{std}}(u) - \sum_k \log \sigma_k$ *with* $\pi_{\mathrm{std}}(u) = \pi(\mu + \sigma \odot u) \prod_k \sigma_k$*.)*

- **Case B2 (joint only).**

  $$\mathrm{LD}_k = \begin{cases} 0, & k = 1, \dots, K-1, \\ \log \pi(x), & k = K, \end{cases} \qquad L = \log \pi(x).$$

  (Maintains shape and row-sum invariants.)

7.4 C) TRTF (Transformation Random Forest; autoregressive triangular CDF map)

Model class. For $k = 1, \dots, K$, estimate conditional CDFs $F_k(y \mid u_{1:k-1})$ and pdfs $f_k(y \mid u_{1:k-1})$ on standardized inputs $u = (x - \mu) \oslash \sigma$.

Fit. For each k: 1. Training tuples $\{(u_{i,1:k-1}, u_{ik})\}_{i \in \mathrm{tr}}$. 2. Grow a transformation forest $\mathcal{T}_k = \{T_{k,t}\}_{t=1}^{T}$.
Each tree partitions $\mathbb{R}^{k-1}$ into leaves ℓ. In leaf ℓ, estimate param $\theta_{k,\ell}$ of a **monotone transformation model** for u_k by maximizing localized log-likelihood

$$\hat{\theta}_{k,\ell} \in \arg\max_{\theta} \sum_{i \in \ell} \log f_k(u_{ik}; \theta),$$

where $f_k(\cdot; \theta)$ is a parametric pdf (e.g. Gaussian/Laplace) and $F_k(\cdot; \theta)$ its CDF.
Splits maximize increase in this objective (with standard complexity controls). 3. Aggregation. For query $u_{1:k-1}$, define weights $w_{k,t,\ell}(u_{1:k-1})$ indicating membership/proximity to leaf ℓ of tree t and set

$$\hat{f}_k(\,\cdot \mid u_{1:k-1}) = \sum_{t,\ell} w_{k,t,\ell}(u_{1:k-1})\, f_k(\,\cdot\,; \hat{\theta}_{k,\ell}),$$

$\hat{F}_k$ analogously. (Enforces monotone CDFs ⇒ invertible in u_k.)

Predict. For each x: 1. $u = (x - \mu) \oslash \sigma$. 2. For $k = 1, \dots, K$: evaluate $\hat{f}_k(u_k \mid u_{1:k-1})$ and set

$$\boxed{\mathrm{LD}_k = \log \hat{f}_k(u_k \mid u_{1:k-1}) - \log \sigma_k,} \qquad L = \sum_k \mathrm{LD}_k.$$

(Equivalently, define a triangular map $\hat{S}_k(u_{1:k}) = \hat{F}_k(u_k \mid u_{1:k-1})$ *to the reference* Unif$(0,1)$*; monotonicity in* u_k *is automatic.)*

7.5 D) TTM-D (Diagonal / Marginal triangular transport)

Parameterization (on standardized u). $S_k(u_k) = a_k + b_k\, u_k, \qquad b_k > 0.$

Training objective (maps from samples; Gaussian reference). $\min_{\{a_k, b_k > 0\}} \sum_{i \in \mathrm{tr}} \sum_{k=1}^{K} \Big(\tfrac{1}{2} S_k(u_{ik})^2 - \log b_k\Big).$

Predict. For each x: $u = (x - \mu) \oslash \sigma, \quad z_k = S_k(u_k), \quad \mathrm{LD}_k = -\tfrac{1}{2} z_k^2 - \tfrac{1}{2}\log(2\pi) + \log b_k - \log \sigma_k, \quad L = \sum_k \mathrm{LD}_k.$

7.6 E) TTM-S (Separable triangular transport)

Parameterization (on standardized u). $S_k(u_{1:k}) = g_k(u_{1:k-1}) + f_k(u_k), \qquad f_k'(u_k) > 0.$ Typical basis: $g_k(u_{1:k-1}) = \sum_j c_{k,j}^{\mathrm{non}}\, \psi_{k,j}^{\mathrm{non}}(u_{1:k-1}),$
$f_k(u_k) = \sum_j c_{k,j}^{\mathrm{mon}}\, \psi_{k,j}^{\mathrm{mon}}(u_k)$ with ψ^{mon} monotone (e.g. iRBF/edge terms).

Training objective (decouples by k). $\min_{\{c_k^{\mathrm{non}}, c_k^{\mathrm{mon}}\}} J_k(c_k^{\mathrm{non}}, c_k^{\mathrm{mon}}) = \sum_{i \in \mathrm{tr}} \Big(\tfrac{1}{2} S_k(u_{i,1:k})^2 - \log f_k'(u_{ik})\Big).$ (Option: eliminate c_k^{non} in closed form given c_k^{mon} via normal equations; then solve a convex box-constrained problem in $c_k^{\mathrm{mon}} \geq 0$.)

Predict. For each x: $u = (x - \mu) \oslash \sigma, \quad z_k = S_k(u_{1:k}), \quad \mathrm{LD}_k = -\tfrac{1}{2} z_k^2 - \tfrac{1}{2}\log(2\pi) + \log f_k'(u_k) - \log \sigma_k, \quad L = \sum_k \mathrm{LD}_k.$

7.7 F) TTM-X (Cross-term triangular transport)

Parameterization (on standardized u). $S_k(u_{1:k}) = g_k(u_{1:k-1}) + \int_0^{u_k} r\big(h_k(t, u_{1:k-1})\big)\, dt, \qquad r : \mathbb{R} \to \mathbb{R}_+$ (e.g. exp, softplus). Here $g_k = \sum_j c_{k,j}^{\mathrm{non}} \psi_{k,j}^{\mathrm{non}}$, and $h_k = \sum_j c_{k,j}^{\mathrm{cr}} \psi_{k,j}^{\mathrm{cr}}$ may include cross-terms $\psi_{k,j}^{\mathrm{cr}}(t, u_{1:k-1})$.

Training objective (per k). $\min_{\{c_k^{\mathrm{non}}, c_k^{\mathrm{cr}}\}} J_k = \sum_{i \in \mathrm{tr}} \Big(\tfrac{1}{2} S_k(u_{i,1:k})^2 - \log \partial_{u_k} S_k(u_{i,1:k})\Big), \quad \partial_{u_k} S_k = r\big(h_k(u_{ik}, u_{i,1:k-1})\big).$ (Compute the integral by 1D quadrature; enforce stability by clipping h_k to $[-H, H]$ during training/inference.)

Predict. For each x: $u = (x - \mu) \oslash \sigma, \quad z_k = S_k(u_{1:k}), \quad \mathrm{LD}_k = -\tfrac{1}{2} z_k^2 - \tfrac{1}{2}\log(2\pi) + \log r\big(h_k(u_k, u_{1:k-1})\big) - \log \sigma_k, \quad L = \sum_k \mathrm{LD}_k.$

7.7.1 Acceptance checks (all models)

- **Shapes.** $\mathrm{LD} \in \mathbb{R}^{N \times K},\ L \in \mathbb{R}^{N},\ \sum_k \mathrm{LD}_{ik} = L_i$ (tol $\leq 10^{-12}$).
- **Constants.** If Gaussian reference is used (TTM-D/S/X), include $-\tfrac{1}{2}\log(2\pi)$ **exactly once per dimension**.

- **Standardization Jacobian.** Subtract $\sum_k \log \sigma_k$ exactly once overall (implemented as $-\log \sigma_k$ inside each LD_k).
- **Determinism.** Same seed $\Rightarrow$ identical outputs (tol $\leq 10^{-15}$).
- **Stability.** No NA/Inf; in TTM-X clip $h_k \in [-H, H]$; for TRTF ensure leaf pdfs bounded away from 0 on support.