Supplementary Material:
Characteristic Circuits

This supplementary document is organized as follows. Appendix A provides a detailed description of notations and the definition of induced trees. Appendix B gives the proof of marginal and moments computation in characteristic circuits. The experimental settings for parameter learning, as well as extra experimental results of numerical integration, are listed in Appendix C. In addition, we provide an analytical solution for calculating the characteristic function distance between two compatible characteristic circuits in Appendix D. Lastly, we describe the statistics of the employed heterogeneous data sets in Appendix E.

A Notation and Background

In this section, we recap the notations and the background.

A.1 Notation

We use the following notations throughout the paper. Let $\mathcal{G}$ be a computational graph with sum nodes S, product nodes P, leaf nodes L, and graph parameters $\theta_{\mathcal{G}}$. Denote $\mathrm{V}(\mathcal{G})$ the vertices of $\mathcal{G}$ and $\mathrm{E}(\mathcal{G})$ the edges of $\mathcal{G}$. The scope of a node N is denoted as $\psi(\mathsf{N})$, with p_{N} the number of RVs in the scope of N. $\mathrm{ch}(\cdot)$ denotes the children of a node.

Let $\boldsymbol{X} = \{X_j\}_{j=1}^{d}$ be a set of random variables. $\boldsymbol{X}$ can also be used as a random vector. Denote $\varphi_{\boldsymbol{X}}(\boldsymbol{t})$ the characteristic function of $\boldsymbol{X}$ for $\boldsymbol{t} \in \mathbb{R}^d$, and $\varphi_{\mathcal{C}}(\boldsymbol{t})$ the estimation of characteristic function from a characteristic circuit $\mathcal{C}$. Let $k \in \mathbb{N}^+$ denote the order of partial derivatives of each variable in $\varphi_{\mathcal{C}}(\boldsymbol{t})$, and $\frac{\partial^{dk}\varphi_{\mathcal{C}}(\boldsymbol{t})}{\partial t_1^k \cdots \partial t_d^k}$ the partial derivative of $\varphi_{\mathcal{C}}(\boldsymbol{t})$ given the order k.

A.2 Induced Trees

The notion of induced trees is proposed to interpret SPNs as deep structured mixture models, which is defined as the following [Zhao et al., 2016].

Definition A.1 (Induced Trees)**.** *Given a complete and decomposable SPN $\mathcal{S}$ over $X = \{X_1, \cdots, X_n\}$, $\mathcal{T} = (\mathcal{T}_V, \mathcal{T}_E)$ is called an induced tree SPN from $\mathcal{S}$ if*

1. *Root($\mathcal{S}$) $\in \mathcal{T}_V$.*
2. *If $v \in \mathcal{T}_V$ is a sum node, then exactly one child of v in $\mathcal{S}$ is in $\mathcal{T}_V$, and the corresponding edge is in $\mathcal{T}_E$.*
3. *If $v \in \mathcal{T}_V$ is a product node, then all the children of v in $\mathcal{S}$ are in $\mathcal{T}_V$, and the corresponding edges are in $\mathcal{T}_E$.*

Here $\mathcal{T}_V$ is the node set of $\mathcal{T}$ and $\mathcal{T}_E$ is the edge set of $\mathcal{T}$.

Given the definition of induced trees, the distribution of an SPN $\mathcal{S}(x)$ can be written as

$$\mathcal{S}(x) = \sum_{i=1}^{\tau} \prod_{(\mathsf{S},\mathsf{N})\in \mathrm{E}(\mathcal{T}_i)} w_{\mathsf{S},\mathsf{N}} \prod_{\mathsf{L}\in \mathrm{V}(\mathcal{T}_i)} p(x \mid \theta_{\mathsf{L}}), \tag{17}$$

where τ denotes the number of induced trees. In the main paper, we use the notion of induced trees to express the inversion output and the moments computation of the characteristic circuit.

B Proofs

B.1 Proof of Marginal Computation

In this subsection, we provide the proof of computing marginals in CCs.

Proof. Let $\mathcal{C} = \langle \mathcal{G}, \psi, \theta_{\mathcal{G}} \rangle$ be a CC on RVs $\boldsymbol{Z} = \boldsymbol{X} \cup \boldsymbol{Y}$ with univariate leave nodes. Further, let $n = |\boldsymbol{X}|$, $m = |\boldsymbol{Y}|$ and let $\boldsymbol{t} = \boldsymbol{t}_{\boldsymbol{X}} \cup \boldsymbol{t}_{\boldsymbol{Y}} \in \mathbb{R}^{n+m}$.

Leaf nodes. For leaf nodes L in $\mathcal{C}$, we have

$$\varphi_{\mathsf{L}}(t_j) = \begin{cases} 1 & \text{if } t_j = 0 \\ \varphi_{\mathsf{L}}(t_j) & \text{otherwise} \end{cases} \tag{18}$$

by definition of CFs.

Product nodes. Without loss of generality, let P be a product node that splits at least one Y_j from its scope into a single child and let this child be denoted as L_j, and the remaining scopes to be $\boldsymbol{X}$. Then by setting $t_j = 0$, we have

$$\varphi_{\mathsf{P}_{\boldsymbol{X}\cup Y_j}}(\boldsymbol{t}_{\boldsymbol{X}} \cup t_j) = \varphi_{\mathsf{P}_{\boldsymbol{X}\cup Y_j}}(\boldsymbol{t}_{\boldsymbol{X}} \cup 0) = \underbrace{\varphi_{\mathsf{L}_j}(0)}_{=1} \prod_{\mathsf{N}\in \mathrm{ch}(\mathsf{P})\setminus \mathsf{L}_j} \varphi_{\mathsf{N}}(\boldsymbol{t}_{\psi(\mathsf{N})}) = \varphi_{\mathsf{P}_{\boldsymbol{X}}}(\boldsymbol{t}_{\boldsymbol{X}}). \tag{19}$$

Sum nodes. We assume sum nodes to be convex combinations, *i.e.*, the weights sum up to one. Therefore, by setting $\boldsymbol{t}_{\boldsymbol{Y}} = 0$ and recursively apply the above, we obtain the sub-circuit corresponding to the marginal of $\boldsymbol{X}$ tractably in CCs. ☐

B.2 Proof of Moments Computation

In this subsection, we provide the proof of computing moments in CCs.

Proof. Let $\mathcal{C} = \langle \mathcal{G}, \psi, \theta_{\mathcal{G}} \rangle$ be a characteristic circuit on RVs $\boldsymbol{X} = \{X_j\}_{j=1}^{d}$ with univariate leave nodes and p_{N} the number of RVs in the scope of N. Denote $k \in \mathbb{N}^{+}$ the order of moments.

Sum Nodes.

Given a sum node S, let $\hat{\boldsymbol{t}} = \boldsymbol{t}_{\psi(\mathsf{S})}$ denote the projection of $\boldsymbol{t}$ onto the scope of S and let $p_{\mathsf{S}} = \|\psi(\mathsf{S})\|_0$ denote the length of the scope of S. Further, let us recall that for smooth sum nodes, the children of the sum node have the same scope, and the scope of the sum node is given as the union of the children's scopes, *i.e.*, equivalent to the scope of each child. Then by linearity, we have:

$$\frac{\partial^{p_{\mathsf{S}}k}\varphi_{\mathsf{S}}(\hat{\boldsymbol{t}})}{\partial \hat{t}_1^k \cdots \partial \hat{t}_{p_{\mathsf{S}}}^k}\Big|_{\hat{t}_1=0,\ldots,\hat{t}_{p_{\mathsf{S}}}=0} = \sum_{\mathsf{N}\in\mathrm{ch}(\mathsf{S})} w_{\mathsf{S},\mathsf{N}} \frac{\partial^{p_{\mathsf{S}}k}\varphi_{\mathsf{N}}(\hat{\boldsymbol{t}})}{\partial \hat{t}_1^k \cdots \partial \hat{t}_{p_{\mathsf{S}}}^k}\Big|_{\hat{t}_1=0,\ldots,\hat{t}_{p_{\mathsf{S}}}=0}, \tag{20}$$

where we applied $\boldsymbol{X}_{\psi(\mathsf{S})} = \boldsymbol{X}_{\psi(\mathsf{N})}$ for $\mathsf{N} \in \mathrm{ch}(\mathsf{S})$ at sum node S. Therefore, computing the derivative at S reduces to a weighted sum of the derivatives at its children.

Product Nodes.

Given a product node P, again let $\hat{\boldsymbol{t}} = \boldsymbol{t}_{\psi(\mathsf{P})}$, and denote $p_{\mathsf{P}} = \|\psi(\mathsf{P})\|_0$ the length of the scope of P.

$$\begin{aligned} \frac{\partial^{p_{\mathsf{P}}k}\varphi_{\mathsf{P}}(\hat{\boldsymbol{t}})}{\partial \hat{t}_1^k \cdots \partial \hat{t}_{p_{\mathsf{P}}}^k}\Big|_{\hat{t}_1=0,\ldots,\hat{t}_{p_{\mathsf{P}}}=0} &= \frac{\partial^{p_{\mathsf{P}}k}\prod_{\mathsf{N}\in\mathrm{ch}(\mathsf{P})}\varphi_{\mathsf{N}}(\boldsymbol{t}_{\psi(\mathsf{N})})}{\partial \hat{t}_1^k \cdots \partial \hat{t}_{p_{\mathsf{P}}}^k}\Big|_{\hat{t}_1=0,\ldots,\hat{t}_{p_{\mathsf{P}}}=0} \\ &= \prod_{\mathsf{N}\in\mathrm{ch}(\mathsf{P})} \frac{\partial^{p_{\mathsf{N}}k}\varphi_{\mathsf{N}}(\boldsymbol{t}_{\psi(\mathsf{N})})}{\partial \hat{t}_1^k \cdots \partial \hat{t}_{p_{\mathsf{N}}}^k}\Big|_{\hat{t}_1=0,\ldots,\hat{t}_{p_{\mathsf{N}}}=0}, \end{aligned} \tag{21}$$

where we utilized $\boldsymbol{X}_{\psi(\mathsf{P})} = \bigcup_{\mathsf{N}\in\mathrm{ch}(\mathsf{P})} \boldsymbol{X}_{\psi(\mathsf{N})}$ and $\bigcap_{\mathsf{N}\in\mathrm{ch}(\mathsf{P})} \boldsymbol{X}_{\psi(\mathsf{N})} = \emptyset$ following the decomposability of product nodes. And in turn we have $\hat{\boldsymbol{t}} = \bigcup_{\mathsf{N}\in\mathrm{ch}(\mathsf{P})} \boldsymbol{t}_{\psi(\mathsf{N})}$ and $\bigcap_{\mathsf{N}\in\mathrm{ch}(\mathsf{P})} \boldsymbol{t}_{\psi(\mathsf{N})} = \emptyset$. Therefore, computing the derivative at P reduces to a product of the derivatives at its children.

Leaf Nodes.

If N is a univariate leaf node L, we can directly have:

$$\frac{\partial^k \varphi_{\mathsf{N}}(t_{\psi(\mathsf{N})})}{\partial t^k_{\psi(\mathsf{N})}}\Big|_{t_{\psi(\mathsf{N})}=0} = \frac{\mathrm{d}^k \varphi_{\mathsf{L}}(t_{\psi(\mathsf{L})})}{\mathrm{d} t^k_{\psi(\mathsf{L})}}\Big|_{t_{\psi(\mathsf{L})}=0}. \tag{22}$$

Through the recursive application of Eq. (20) and Eq. (21), we obtain that Eq. (15) reduces to derivatives at the leaves and can be computed efficiently. □

Note that the moments computation can be easily extended to the mixed moments, where each random variable can have a different order of derivatives.

C Experiments

C.1 Experimental Settings for Parameter Learning

In this section, we illustrate the details of the settings for parameter learning.

Random Structure. The random structure for parameter learning is created by recursively creating sum and product nodes. A sum node is created with random normalised weights with two children, and a product node is created by randomly splitting the scopes into two subsets. The splitting terminates when there is only one scope at a node, and then a leaf node with randomly initialised parameters is created. The categorical leaf nodes are initialised with uniformly sampled weights after normalization, the mean and location of normal and α-stable distribution leaves are initialised with samples from a normal distribution centred at the average of training data. Gaussian leaves are used for synthetic data sets and α-stable distribution leaves are used for the UCI data sets.

Parameter Learning. For all parameter learning experiments, we employ a linearly decreasing learning rate from lr_1 to lr_2 with iterations $iter$. The gradient is obtained from the training data without using batches, as the data sets in our experiments are of small size. The objective CFD is calculated with a fixed $\eta = 1$ and $k = 100$.

For results on synthetic data sets in Table 1 from the main paper, the column *Random Structure* is directly evaluated from the above randomly initialised CC without parameter learning. The results of *Random Structure & Parameter Learning* are obtained from the above CC after parameter learning with $lr_1 = 0.5$, $lr_2 = 0.01$ and $iter = 300$ for data set `MM`, and $lr_1 = 1.0$, $lr_2 = 0.05$ and $iter = 40$ for data set `BN`. *Structure Learning* follows the structure learning setup described in the main body. When applying parameter learning on the model from *Structure Learning* with $lr_1 = 0.5$, $lr_2 = 0.005$ and $iter = 300$ for data set `MM`, and $lr_1 = 0.5$, $lr_2 = 0.01$ and $iter = 200$ for data set `BN`, we obtain the results of *Structure Learning & Parameter Learning*. Finally, we randomise the weights and leaf parameters of the model from *Structure Learning* and apply parameter learning with $lr_1 = 0.5$, $lr_2 = 0.01$ and $iter = 300$ for data set `MM`, and $lr_1 = 1.0$, $lr_2 = 0.05$ and $iter = 40$ for data set `BN`, resulting in *Structure Learning (random* $\mathbf{w}$*) & Parameter Learning*. Note that the mean of a Gaussian leaf is initialised with samples from a normal distribution centred at the average of training data.

C.2 Numerical Integration with Quadrature

Throughout the paper, we use Gauss-Hermit quadrature for numerical integration at the leaves. In order to demonstrate the reliability of the numerical integration, we test with an increasing number of grid points $\{50, 100, 200, 300\}$ through quadrature and show the corresponding output at the root in Fig. 5. The CCs are learned from structure learning with either the simplified G-test-based splitting or the random dependency coefficient (RDC) based splitting. For the RDC-based splitting, the threshold, denoted as ξ, is chosen from grid search from $\{0.1, 0.2, \cdots, 0.9\}$ on each validation set with 50 grid points for the quadrature. The results indicate that a low value of degree in quadrature is sufficient since numerical integration is only required on the real line (1D). The results also show that RDC-based structure learning outperforms the G-test-based splitting on most of the data sets, as it is designed based on an independence test on heterogeneous data.

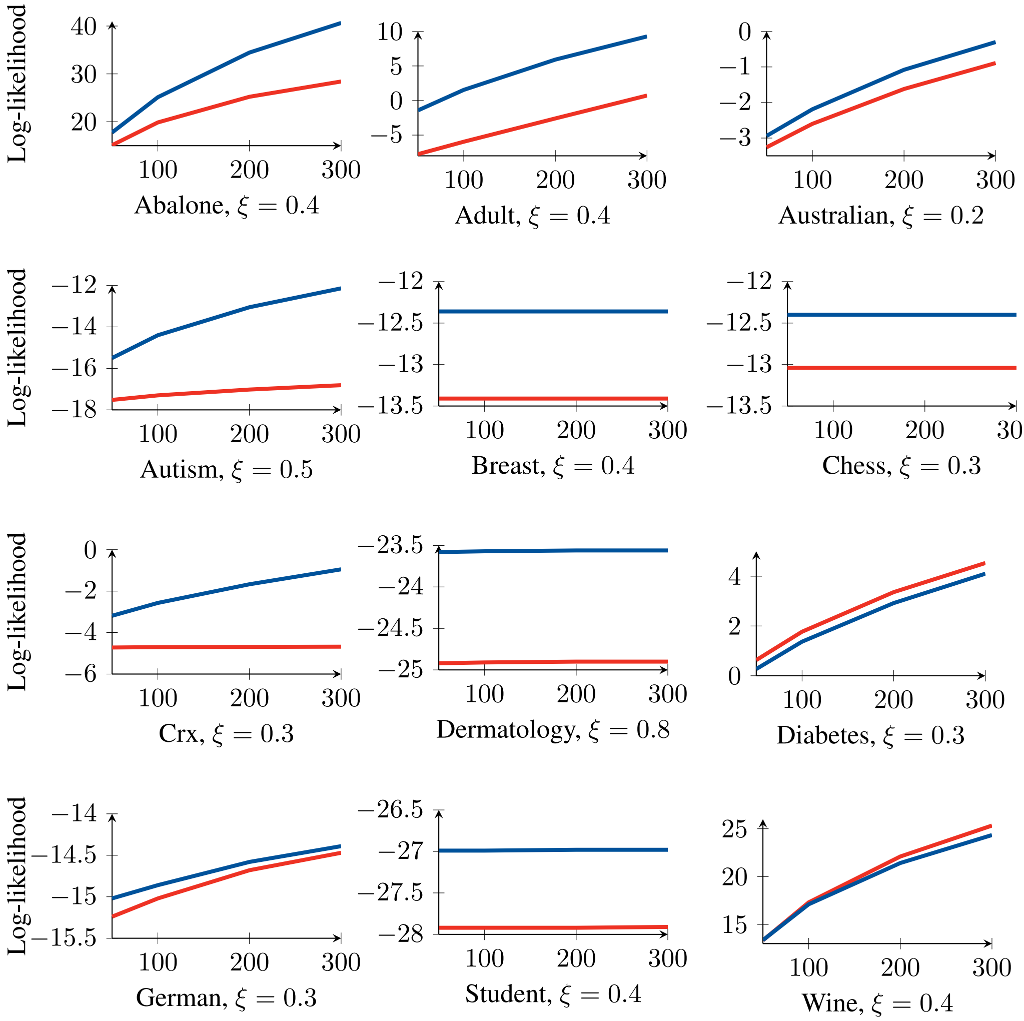

Figure 5: Log-likelihoods from CCs with varying numbers of grid points in the quadrature (x-axis). The CCs are learned from structure learning with either simplified G-test 🟥 or RDC 🟦 based splitting for product nodes.

D Analytical Solution of the Characteristic Function Distance

The squared characteristic function distance (CFD)

$$\mathrm{CFD}^2_\omega(\mathbb{P},\mathbb{Q}) = \int_{\mathbb{R}^d} \left|\varphi_\mathbb{P}(\boldsymbol{t}) - \varphi_\mathbb{Q}(\boldsymbol{t})\right|^2 \omega(\boldsymbol{t};\eta)\mathrm{d}\boldsymbol{t} \tag{23}$$

can not only be estimated with MC methods by sampling from $\omega(\boldsymbol{t};\eta)$, but also be calculated through the characteristic circuits analytically, if $\varphi_\mathbb{P}(\boldsymbol{t})$ and $\varphi_\mathbb{Q}(\boldsymbol{t})$ are compatible characteristic circuits.

Eq. (23) can be rewritten as

$$\begin{aligned}\mathrm{CFD}^2_\omega(\mathbb{P},\mathbb{Q}) &= \int_{\mathbb{R}^d} \left(\varphi_\mathbb{P}(\boldsymbol{t}) - \varphi_\mathbb{Q}(\boldsymbol{t})\right)\left(\overline{\varphi_\mathbb{P}(\boldsymbol{t}) - \varphi_\mathbb{Q}(\boldsymbol{t})}\right)\omega(\boldsymbol{t};\eta)\mathrm{d}\boldsymbol{t} && (24)\\ &= \int_{\mathbb{R}^d} \left(\varphi_\mathbb{P}(\boldsymbol{t})\overline{\varphi_\mathbb{P}(\boldsymbol{t})} - \varphi_\mathbb{P}(\boldsymbol{t})\overline{\varphi_\mathbb{Q}(\boldsymbol{t})} - \varphi_\mathbb{Q}(\boldsymbol{t})\overline{\varphi_\mathbb{P}(\boldsymbol{t})} + \varphi_\mathbb{Q}(\boldsymbol{t})\overline{\varphi_\mathbb{Q}(\boldsymbol{t})}\right)\omega(\boldsymbol{t};\eta)\mathrm{d}\boldsymbol{t}, && (25)\end{aligned}$$

where $\overline{z}$ denotes the conjugate of the complex number z. Without loss of generality, let us derive the analytical solution of $\int_{\mathbb{R}^d} \varphi_\mathbb{P}(\boldsymbol{t})\overline{\varphi_\mathbb{Q}(\boldsymbol{t})}\omega(\boldsymbol{t};\eta)\mathrm{d}\boldsymbol{t}$, since the derivation can be directly applied to the other terms in Eq. (25). In the following, we omit the term $\omega(\boldsymbol{t};\eta)$ at sum and product nodes for simplicity. At sum nodes S and S',

$$\int \mathsf{S}(\boldsymbol{t})\overline{\mathsf{S}'(\boldsymbol{t})}\mathrm{d}\boldsymbol{t} = \int \left(\sum_{\mathsf{N}\in\mathrm{ch}(\mathsf{S})} w_{\mathsf{S},\mathsf{N}}\mathsf{N}(\boldsymbol{t})\right)\left(\sum_{\mathsf{N}'\in\mathrm{ch}(\mathsf{S}')} w_{\mathsf{S}',\mathsf{N}'}\overline{\mathsf{N}'(\boldsymbol{t})}\right)\mathrm{d}\boldsymbol{t} \tag{26}$$

$$= \int \sum_{\mathsf{N}\in\mathrm{ch}(\mathsf{S})}\sum_{\mathsf{N}'\in\mathrm{ch}(\mathsf{S}')} w_{\mathsf{S},\mathsf{N}}w_{\mathsf{S}',\mathsf{N}'}\mathsf{N}(\boldsymbol{t})\overline{\mathsf{N}'(\boldsymbol{t})}\mathrm{d}\boldsymbol{t} \tag{27}$$

$$= \sum_{\mathsf{N}\in\mathrm{ch}(\mathsf{S})}\sum_{\mathsf{N}'\in\mathrm{ch}(\mathsf{S}')} w_{\mathsf{S},\mathsf{N}}w_{\mathsf{S}',\mathsf{N}'}\int \mathsf{N}(\boldsymbol{t})\overline{\mathsf{N}'(\boldsymbol{t})}\mathrm{d}\boldsymbol{t}. \tag{28}$$

At product nodes P and P',

$$\int \mathsf{P}(\boldsymbol{t})\overline{\mathsf{P}'(\boldsymbol{t})}\mathrm{d}\boldsymbol{t} = \int \left(\prod_{\mathsf{N}\in\mathrm{ch}(\mathsf{P})} \mathsf{N}(\boldsymbol{t}_{[\psi(\mathsf{N})]})\right)\left(\prod_{\mathsf{N}'\in\mathrm{ch}(\mathsf{P}')} \overline{\mathsf{N}'(\boldsymbol{t}_{[\psi(\mathsf{N}')]})}\right)\mathrm{d}\boldsymbol{t} \tag{29}$$

$$= \int \prod_{(\mathsf{N},\mathsf{N}')\in\Delta_{\mathsf{P}\times\mathsf{P}'}} \mathsf{N}(\underbrace{\boldsymbol{t}_{[\psi(\mathsf{N})]}}_{=\hat{\boldsymbol{t}}})\overline{\mathsf{N}'(\boldsymbol{t}_{[\psi(\mathsf{N}')]})}\,\mathrm{d}\boldsymbol{t} \qquad \text{(compatibility)} \tag{30}$$

where $\Delta_{\mathsf{P}\times\mathsf{P}'}$ denotes the diagonal of the Cartesian product of the children of P and P', *i.e.*, $\mathrm{diag}(\mathrm{ch}(\mathsf{P}) \times \mathrm{ch}(\mathsf{P}'))$, compatibility ensures that both product nodes apply the same partition of the scope $\psi(\mathsf{P}) = \psi(\mathsf{P}')$ with parts in the same order, and $\boldsymbol{t}_{[\psi(\mathsf{N})]}$ is the projection of $\boldsymbol{t}$ to the scope of N. Therefore,

$$= \prod_{(\mathsf{N},\mathsf{N}')\in\Delta_{\mathsf{P}\times\mathsf{P}'}} \int_{\mathbb{R}^{p_{\mathsf{N}}}} \mathsf{N}(\hat{\boldsymbol{t}})\overline{\mathsf{N}'(\hat{\boldsymbol{t}})}\,\mathrm{d}\hat{\boldsymbol{t}}. \qquad \text{(compatibility)} \tag{31}$$

At univariate leaf nodes L and L', assuming both leaf nodes model univariate normal distribution with parameters (μ,σ) and (μ',σ'), and $\omega(t;\eta) = \frac{1}{\eta\sqrt{2\pi}}\exp(\frac{-t^2}{2\eta^2})$, then

$$\int_{\mathbb{R}} \mathsf{L}(t)\overline{\mathsf{L}'(t)}\omega(t;\eta)\mathrm{d}t = \int_{\mathbb{R}} \exp(\mathrm{i}\,t\,\mu - \frac{1}{2}\sigma^2t^2)\overline{\exp(\mathrm{i}\,t\,\mu' - \frac{1}{2}\sigma'^2t^2)}\frac{1}{\eta\sqrt{2\pi}}\exp(\frac{-t^2}{2\eta^2})\mathrm{d}t \tag{32}$$

$$= \frac{1}{\eta\sqrt{2\pi}}\int_{\mathbb{R}} \exp(\mathrm{i}\,t\,(\mu-\mu') - \frac{1}{2}(\sigma^2+\sigma'^2+\frac{1}{\eta^2})t^2)\mathrm{d}t \qquad (\overline{e^z} = e^{\overline{z}}) \tag{33}$$

$$= \frac{1}{\eta\hat{\sigma}}\exp(\frac{-\hat{\mu}^2}{2\hat{\sigma}^2}), \qquad \text{(integral of a Gaussian function)} \tag{34}$$

where $\hat{\mu} = \mu - \mu'$ and $\hat{\sigma} = \sqrt{\sigma^2 + \sigma'^2 + 1/\eta^2}$. Therefore, at univariate leaf nodes, it can be solved either analytically or with Monte-Carlo integration: $\int_{\mathbb{R}} \mathsf{L}(t)\overline{\mathsf{L}'(t)}\omega(t;\eta)\mathrm{d}t \approx \frac{1}{k}\sum_{j=1}^{k}\varphi_{\mathsf{L}}(t_j)\overline{\varphi_{\mathsf{L}'}(t_j)}$, where $\{t_1,\cdots,t_k\} \overset{\text{i.i.d.}}{\sim} \omega(t;\eta)$. With the above properties, the CFD between two compatible CCs can be calculated from the bottom-up analytically and efficiently.

E Statistics of the Heterogeneous Data Sets

In this section we briefly describe some statistics of the heterogeneous data sets, to provide a better and more detailed view of the data sets.

Table 3: Statistics of the heterogeneous data sets, including the number of instances in the training/validation/test sets and the number of total/discrete/continuous RVs for each subset.

Data Set	#train	#val	#test	#RVs	#D. RVs	#C. RVs
Abalone	2923	418	836	9	1	8
Adult	22792	3256	6513	13	7	6
Australian	482	70	138	10	3	7
Autism	2464	352	705	25	15	10
Breast	476	68	137	10	9	1
Chess	19639	2805	5612	7	7	0
Crx	455	65	131	11	5	6
Dermatology	256	36	74	34	33	1
Diabetes	537	77	154	8	1	7
German	700	99	201	17	14	3
Student	276	40	79	20	19	1
Wine	4547	650	1300	12	1	11